

Exhibit A

**IN THE UNITED STATES DISTRICT COURT
FOR THE
NORTHERN DISTRICT OF ILLINOIS**

CASCADES COMPUTER INNOVATION, LLC,

Plaintiff(s),

v.

SAMSUNG ELECTRONICS CO., LTD.,

Defendant(s).

Case No. 11 C 4574

Judge Matthew F. Kennelly

JUDGMENT IN A CIVIL CASE

Judgment is hereby entered (check appropriate box):

☐ in favor of plaintiff(s)
and against defendant(s)
in the amount of \$ _____,

which ☐ includes pre-judgment interest.
☐ does not include pre-judgment interest.

Post-judgment interest accrues on that amount at the rate provided by law from the date of this judgment.

Plaintiff(s) shall recover costs from defendant(s).

☐ in favor of defendant(s)
and against plaintiff(s)

Defendant(s) shall recover costs from plaintiff(s).

☒ other: Judgment is entered on the jury's verdict as follows:

Claims 1 and 15 of U.S. Patent No. 7,065,750 are not invalid. Defendant Samsung Electronics Co., Ltd. does not infringe claims 1 and 15 of U.S. Patent No. 7,065,750. Plaintiff Cascades Computer Innovation, LLC takes nothing from defendant Samsung Electronics Co., Ltd.

This action was (*check one*):

☒ tried by a jury with Judge Matthew F. Kennelly presiding, and the jury has rendered a verdict.
☐ tried by Judge _____ without a jury and the above decision was reached.
☐ decided by Judge _____ on a motion _____.

Date: 9/21/2015

Thomas G. Bruton, Clerk of Court

Pamela J. Geringer, Deputy Clerk

Exhibit B

**IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF ILLINOIS
EASTERN DIVISION**

**CASCADES COMPUTER INNOVATION,
LLC,**

Plaintiff,

VS.

SAMSUNG ELECTRONICS CO. LTD.,

Defendant.

No. 11 C 4574

**CASCADES COMPUTER INNOVATION,
LLC,**

Plaintiff,

VS.

HTC CORPORATION,

Defendant.

No. 11 C 6235

MEMORANDUM OPINION AND ORDER

MATTHEW F. KENNELLY, District Judge:

Under the doctrine of patent exhaustion, the initial authorized sale of a patented item terminates the patent holder's rights to that item, and the patent holder may not sue a downstream user of the item for infringement. Patent exhaustion also applies to method patents. If the holder of a method patent authorizes another to practice the patented method, the patent holder cannot successfully sue for infringement those who acquire from the authorized user a product that substantially embodies the patented

method.

In these cases, Cascades Computer Innovation LLC has sued Samsung Electronics Co. Ltd. and HTC Corporation for infringement of U.S. Patent 7,065,750 (the '750 patent). Cascades contends that defendants manufacture and sell products that practice the method claimed in the patent. Samsung and HTC have moved for summary judgment. They contend that Cascades authorized Google to practice the patented method in its Android operating system and that their claimed infringement arises from their use of that same operating system, which they acquired from Google. Thus, Samsung and HTC contend, the doctrine of patent exhaustion bars Cascades's claims against them.

Background

Because the defendants have moved for summary judgment, the Court "constru[es] all facts and reasonable inferences in the light most favorable to the non-moving party," in this case, Cascades. *Ellis v. DHL Exp. Inc.*, 633 F.3d 522, 525 (7th Cir. 2011).

The '750 patent is entitled "Method and Apparatus for Preserving Precise Exceptions in Binary Translated Code." Pl.'s Third Am. Compl. ¶ 11. In general terms, it describes a method for efficiently executing on one system architecture computer programming code that is intended for a different architecture. Until 2014, Cascades was an exclusive licensee under the '750 patent, with an exclusive right to sue for the patent's past, present, and future infringement.

In 2011, Cascades filed patent infringement suits against certain parties, including Samsung and HTC. Cascades contends that Samsung and HTC infringe the

'750 patent by manufacturing and selling smartphones and tablets that use the Dalvik JIT Compiler, which is part of the Android operating system distributed by Google. According to Cascades, "[t]he claimed method is performed when a user of the cellular phones operates the device for their intended purpose using the Android operating system, e.g., allowing the Dalvik Virtual Machine to optimize the byte code for each application." *Id.* ¶ 14.

On January 29, 2014, Cascades entered into a settlement and license agreement with Google, the parent of Motorola Mobility LLC, previously a defendant in one of Cascades's suits. In exchange for a one-time fee, Cascades granted Google:

a worldwide, non-exclusive, fully paid up and perpetual and irrevocable license under Cascades Patents to practice and undertake any of and all of the rights granted a patent owner under 35 U.S.C. 101, et seq., and under their counterparts, under the laws of foreign jurisdictions including, but not limited to, the right to make, have made, use, sell, offer to sell, export, import, and otherwise practice and/or have practiced for Google or a Google Affiliate, any and all claims of the Cascades Patents *in any Google Product*.

Defs.' Ex. 1 at 2 (emphasis added). The license agreement provides that the term "Google Products" includes "products of Google, Motorola and/or Google Affiliates, including all Motorola and Nexus devices, but . . . excludes mobile devices manufactured by third parties and running the Android OS except any Nexus-branded devices." *Id.*

As indicated earlier, Cascades's claims of infringement focus on the use of a feature of the Android operating system called the Dalvik JIT Compiler. Cascades has identified no other feature of the defendants' devices or the operating system they use that infringes the '750 patent. The quoted term of the license agreement between Cascades and Google entitles Google to use, sell, or practice the patented method in

any "Google Product." It is undisputed that the Android operating system is a Google product; no reasonable fact finder could find otherwise. The same is true of the Dalvik JIT Compiler. Thus the license agreement authorized Google, from that day forward, to convey the Android operating system—including the Dalvik JIT Compiler—without fear of a claim of infringement by Cascades. The agreement's definition of "Google products," however, purported to limit this to certain types of devices, *not* including those made by Samsung and HTC.

The settlement and license agreement between Cascades and Google also included a release and a covenant not to sue. The release provides that Cascades

releases and discharges Google, Motorola, Google Affiliates, Google Partners . . . from any and all claims, demands, debts, liabilities, actions, causes of actions or suits of whatever kind of nature, asserted or not asserted, known or unknown, arising out of the claims or matters that have been or could have been asserted by Cascades in the Actions relating to the same facts and circumstances therein, provided that such release and discharge shall not extend to any other defendant in the Actions.

Defs.' Ex. 1 at 4. The covenant not to sue states that

Cascades covenants not to sue Google, Motorola, Google Affiliates and/or Google Partners for any infringement or any other violation of the Cascades Patents based upon any licensed activity, permitted pursuant to this Section, related to any Google Product; provided this covenant does not extend to any other defendant in the Patent Suit.

Id. at 3.

Google provides the Android open source code to all sorts of device manufacturers, including Samsung and HTC. Samsung and HTC make and sell devices that use the Android operating system. As indicated, that operating system embodies the allegedly infringing Dalvik JIT Compiler.

Samsung and HTC contend that by virtue of the license agreement between

Cascades and Google, the doctrine of patent exhaustion bars Cascades from pursuing its patent infringement claims against them. They contend that despite the license agreement's limitations regarding its scope, Cascades's grant to Google of a license to convey the Android operating system to others enables those who so acquire the operating system to use it as they wish, without risk of liability for infringement of the '750 patent. Samsung and HTC contend that Cascades's agreement with Google—specifically, the release and/or the covenant not to sue—also bars Cascades from suing them for past infringement.

Discussion

Summary judgment is appropriate "if the movant shows that there is no genuine dispute as to any material fact and the movant is entitled to judgment as a matter of law." Fed. R. Civ. P. 56(a). "Patent exhaustion is an affirmative defense to a claim of patent infringement . . . and like other issues in which there are no disputed factual questions, may be properly decided on summary judgment." *Keurig, Inc. v. Sturm, Foods, Inc.*, 732 F.3d 1370, 1373 (Fed. Cir. 2013).

As the Court has indicated, the doctrine of patent exhaustion "provides that the initial authorized sale of a patented item terminates all patent rights to that item." *Quanta Computer Inc. v. LG Elecs., Inc.*, 553 U.S. 617, 625 (2009). The rationale behind this doctrine is that the sale of a patented device "exhausts the patentee's right to control the purchaser's use of that item . . . because the patentee has bargained for and received full value for the goods." *Keurig*, 732 F.3d at 1373. In short, the doctrine of patent exhaustion prevents a patent holder from benefitting multiple times from a single conveyance by claiming that downstream users infringe the patent when they use

the item they acquired.

Application of the doctrine of patent exhaustion does not depend on the existence of a sale for consideration. In *Lifescan Scotland, Ltd. v. Shasta Technologies, LLC*, 734 F.3d 1361, 1377 (Fed. Cir. 2013), the Federal Circuit concluded that patent exhaustion applied even though plaintiff had given away the items at issue rather than sold them. The court noted that "in the case of an authorized and unconditional transfer of title, the absence of consideration is no barrier to the application of patent exhaustion principles." *Id.* at 1374. Thus the fact that Google gives away the Android operating system is no barrier to application of the doctrine of patent exhaustion.

Patent exhaustion applies to patented methods just as it applies to patented devices. *Quanta*, 553 U.S. at 621; *United States v. Univis Lens Co.*, 316 U.S. 241, 250-51 (1942). A method claim is exhausted by an authorized sale of an item—even an unpatented item—"that substantially embodies the [patented] method if the item (1) has no reasonable noninfringing use and (2) includes all inventive aspects of the claimed method." *Keurig*, 732 F.3d at 1373. And "alternative uses are relevant to the exhaustion inquiry under *Quanta* only if they are both reasonable *and intended* by the patentee or its authorized licensee." *Lifescan Scotland*, 734 F.3d at 1369 (internal quotation marks omitted).

A. Effect of the limitations in the license agreement

As the Court has indicated, Samsung and HTC argue that via the settlement and license agreement, Cascades gave Google the right to convey the Android operating system (including the Dalvik JIT Compiler) to others, leaving Google free to provide the technology to whomever it chose, including Cascades and HTC. Cascades and HTC

argue that they were then free to use the technology in whichever way *they* chose, free from any claim of infringement of Cascades's patent.

In response, Cascades relies on the principle that "[e]xhaustion is triggered only by a sale authorized by the patent holder." *Quanta*, 553 U.S. at 636 (citing *Univis*, 316 U.S. at 249). It argues that the plain language of the license agreement clearly "excludes Android as it is used in Defendants' devices." Pl.'s Resp. Br. at 5. Cascades argues, in other words, that the conveyance of the Android operating system by Google to Samsung and HTC was not an "authorized" sale that exhausted Cascades's patent rights in the operating system. Specifically, Cascades says, "[e]ven accepting for the sake of argument that Android is a product made by Google, the definition of 'Products' in the Motorola Agreement was modified by the definition of 'Google Product'. . . ." *Id.* at 6. Samsung and HTC maintain that the agreement's exclusion of their products is ineffective, because "Cascades' license to Google creates *exhaustion* with respect to Defendants' downstream products that incorporate Google's licensed products." Defs.' Opening Br. at 10.

The Supreme Court addressed a similar scenario in *Quanta*. In that case, LG Electronics (LGE) held rights under patents describing, among other things, methods for accessing and transferring data within a computer. Pursuant to a license agreement, LGE gave Intel the right to manufacture and sell microprocessors practicing these methods. But the agreement contained express limitations; specifically, it did not extend to any product made by combining an Intel product with a non-Intel product. Intel exercised its rights under the agreement by selling microprocessors to Quanta, which proceeded to use them in computers combining Intel and non-Intel products.

LGE argued that this was not an authorized sale and thus patent exhaustion did not apply.

The Court rejected Intel's argument. The license, the Court observed, did not limit Intel's ability to sell products that practiced the patents. "Because Intel was authorized to sell . . . products [that practiced the LGE patents] to Quanta," the Court concluded, "the doctrine of patent exhaustion prevents LGE from further asserting its patent rights with respect to patents substantially embodied by those products."

Quanta, 553 U.S. at 637. *Quanta* built on the longstanding principle that "the sale by a person who has the full right to make, sell, and use" an item (in *Quanta*, Intel), "carries with it the right to the use of that machine to the full extent to which it can be used."

Adams v. Burke, 84 U.S. 453, 455 (1873) (concluding that the purchaser of a coffin from a company that held a right to manufacture, sell, and use certain coffin lids had the right to use that coffin for the purpose for which all coffins are used, even though the

company's right was limited to a certain geographic radius). *See also Helferich Patent Licensing, LLC v. New York Times Co.*, 965 F. Supp. 2d 971, 978 (N.D. Ill. 2013)

(internal quotation marks omitted) ("[O]nce lawfully made and sold, there is no restriction on [the product's] use to be implied for the benefit of the patentee or his assignee or licensees.").

By way of its license agreement with Cascades, Google was authorized to convey to others, including Samsung and HTC, products—including the Android operating system—that practiced Cascades's patents. As a result, Cascades could no longer assert patent rights with respect to those products. As was the case in *Quanta*, use of the restriction in the Cascades/Google license agreement to limit how those who

thereafter acquired the Android operating system from Google could use it would in effect allow Cascades to circumvent the patent exhaustion doctrine and reap multiple gains from a single sale. See *Keurig*, 732 F.3d at 1374. The license authorized Google to convey the Android operating system to others, and thus the conveyance of the operating system to Samsung and HTC was an authorized sale. The agreement's attempt to carve out downstream users' own mobile devices is ineffective under *Quanta*.

For these reasons, the Court concludes that Samsung and HTC have established that there was an authorized sale of the Android operating system (and the Dalvik JIT Compiler) to them; no reasonable fact-finder could determine otherwise. The Court therefore proceeds to consider whether Samsung and HTC have met the remaining requirements for patent exhaustion.

B. Substantial embodiment and reasonable noninfringing uses

As the Court has indicated, to establish their patent exhaustion defense, Samsung and HTC must show that the Android operating system as it was conveyed to them "substantially embodies" Cascades's patented method in that the operating system "(1) has no reasonable noninfringing use and (2) includes all inventive aspects of the claimed method." *Keurig*, 732 F.3d at 1373.

Cascades maintains that the Android operating system does not substantially embody the '750 patent because it has a reasonable noninfringing use. It argues that "[t]he Kit Kat version of Android [which Samsung purportedly uses in its devices] does not require the use of the JIT compiler and instead allows a user to switch runtimes from the JIT default to ART (Android Run Time)." Pl.'s Resp. Br. at 6. Cascades also says that one can "remove the claimed inventions from the Accused Devices" by

"redesign[ing] and modify[ing] the JIT Compiler to use a Kelly-style method of handling precision exceptions." *Id.* at 10. Cascades says these should be considered intended uses, because "the Accused Devices . . . would likely still function with unchanged performance" *Id.* (internal quotation marks omitted).

The question, however, is whether there is a reasonable alternative noninfringing use of the Dalvik JIT Compiler—itself a "Google product" under the Cascades-Google agreement—not whether defendants could avoid liability by enabling its non-use. As defendants argue, this argument is foreclosed by *Quanta*, in which the Court noted that evidence that patented features of certain products could be disabled did not show a reasonable noninfringing use of those features, because "[t]he disabled features would have no real *use*." *Quanta*, 553 U.S. at 632 n.6.

That aside, Cascades has offered no admissible evidence from which a reasonable finding could be made that a user's ability to enable Android Run Time rather than the Dalvik JIT Compiler constitutes a *reasonable* noninfringing use. One of Cascades's experts reports that "the official Android website warns its KitKat users about the risks of running ART: **Important:** *Dalvik must remain the default runtime or your risk breaking your Android implementations and third-party applications.*" Defs.' Ex. 21 ¶ 141. The same expert reports that "Android suggests that Dalvik remains as the default runtime against the risk of malfunctioning regarding Android implementations." *Id.* ¶ 52 (internal quotation marks omitted). It is difficult to see how purported noninfringing uses that run this sort of a risk could be considered reasonable.

For these reasons, the Court concludes that defendants have established that the Cascades-Google license brings the doctrine of patent exhaustion into play; no

reasonable fact finder could determine otherwise. Defendants are therefore entitled to summary judgment on infringement from the date of the Cascades-Google agreement forward.

C. Past infringement

Defendants also argue that the release and/or covenant not to sue in the Cascades-Google agreement exhausts Cascades' claims against them for infringement predating the agreement. Defendants' theory is that "[b]ecause Google was released for any past infringement, any past distribution of its Android software is 'authorized,' thereby triggering the patent exhaustion doctrine." Defs.' Reply at 14. Cascades argues that "the alleged exhaustion of Plaintiff's patent rights do [sic] not extend backwards to the first sale of Android" because a license is needed to bring that about and a release is insufficient. Pl.'s Resp. Br. at 13.

The Court concludes that Cascades has the better of this argument. Patent exhaustion is a rule that looks forward; it results from the patent holder's sale of a patented item or a product embodying a patented method, or a grant of a license permitting another to use the patented item or method. A release, by contrast, looks backward and takes an alleged infringer off the hook for something it has already done or is alleged to have done. See, e.g., *Waterloo Furniture Components Ltd. v. Haworth*, 467 F.3d 641, 647 (7th Cir. 2006); *Ransburg Electro-Coating Corp. v. Spiller & Spiller, Inc.*, 489 F.2d 974, 977 (7th Cir. 1973) ("[A] release for past wrongdoing is not the equivalent of a license to do rightfully the same thing in the future.") (citing, among other cases, *Rude v. Wescott*, 180 U.S. 152 (1889)).

Perhaps as importantly, the release and covenant not to sue in this case were

expressly limited to Google and its affiliated entities. As indicated earlier, when a patent holder sells or grants a license on a patented item or method, the patent holder surrenders its rights to enforce the patent against not only the buyer or licensee but also those who acquire the product or method from the buyer or licensee. A release, however, works differently. Specifically, a release given to one tortfeasor does not release others and does not, unlike a sale or a license, surrender the releasor's rights. See, e.g., *Glenayre Elecs., Inc. v. Jackson*, 443 F.3d 851, 868 (Fed. Cir. 2006) (citing *Aro Mfg. Co. v. Convertible Top Replacement Co.*, 377 U.S. 476, 501 (1964)); see also *Holland v. United States*, 621 F.3d 1366, 1380 (Fed. Cir. 2010) (release of one jointly responsible party does not release others); *Leung v. Verdugo Hills Hosp.*, 55 Cal. 4th 291, 301-02, 282 P.3d 1250, 1255-56 (2012) (California law, which applies under the Cascades-Google agreement; release of one tortfeasor does not release others). There is no case from the Supreme Court or the Federal Circuit—at least defendants cite none—that suggests that limitations in a release that restrict its operation to a certain party or parties cannot be enforced.

Defendants contend that a covenant not to sue operates the same way as a license for purposes of the doctrine of patent exhaustion. They rely on *TransCore, LP v. Elec. Transaction Consultants Corp.*, 563 F.3d 1271 (Fed. Cir. 2009). In *TransCore*, the Court held that a settlement agreement that included a covenant not to sue for all future infringement claims authorized the party to sell patented items, and thus exhausted plaintiff's rights in those items with respect to future sales. *Id.* at 1276. This case does not support defendants' position regarding past infringement. The covenant in *TransCore* expressly concerned future infringing activity. That made it essentially

indistinguishable from a license authorizing a party to use a patented item or method. Nothing in *TransCore* or, as best as the Court can determine, any other Federal Circuit case, says that the same is true of a release or covenant like those in the present case that, by their terms, operate retrospectively and expressly carve out non-parties.

Defendants have cited two district court decisions that they contend support their position. In *PSN Illinois LLC v. Abbott Labs*, No. 09 C 5879, 2011 WL 4442825 (N.D. Ill. Sept. 20, 2011), the court appears to have read *TransCore* as supporting the proposition that a backward-looking release invokes the doctrine of patent exhaustion for downstream users just as a forward-looking sale or license does. The Court respectfully disagrees with this decision; it does not take account of the fact that *TransCore* concerned sales postdating the covenant not to sue in that case, which expressly covered future sales. *Bobel v. Maxlite, Inc.*, No. 12 C 5346, 2013 WL 142987 (N.D. Ill. Jan. 11, 2013), the other case defendants cite, is harder to interpret because the decision is heavily redacted. Thus it is unclear what the covenant not to sue in that case said or exactly when the sales in question had occurred vis-à-vis the giving of the covenant. But to the extent the decision in *Bobel* reads *TransCore* as defendants suggest, the Court respectfully disagrees with it for the reasons discussed.

Conclusion

For the foregoing reasons, the Court grants defendants' motion for summary judgment in part and denies it in part [Case No. 11 C 4574, dkt. nos. 156 & 156; Case No. 11 C 6235, dkt. nos. 123 & 124]. The Court previously granted HTC's motion to supplement its summary judgment motion [Case No. 11 C 6235, dkt. no. 141], so that motion is hereby terminated. Both cases remain set for a status hearing on September

30, 2014, but the time of the status hearing is advanced to 9:00 a.m. on that date, and it will be conducted by telephone. Given the number of attorneys involved, counsel are directed to make appropriate arrangements for a call-in number and are to communicate those arrangements to the undersigned judge's chambers at least one business day prior to the call.


MATTHEW F. KENNELLY
United States District Judge

Date: September 14, 2014

Exhibit C

**UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF ILLINOIS
EASTERN DIVISION**

Cascades Computer Innovation, LLC,

Plaintiffs,

v.

HTC Corporation

Defendants.

Civil Action No.: 1:11-cv-6235

The Hon. Matthew F. Kennelly

JOINT MOTION TO STAY PROCEEDINGS INVOLVING HTC

Plaintiff, Cascades Computer Innovation LLC (“Cascades”), and Defendant, HTC Corporation (“HTC”), respectfully request the Court to stay all proceedings (including the trial currently scheduled to begin on September 14, 2015) for the above captioned matter.

On September 7, 2011, Cascades filed a complaint against HTC Corporation (“HTC”) alleging the infringement of United States Patent No. 7,065,750 (the “’750 patent”). [D.I. 1]. Cascades had filed an earlier complaint against Samsung Electronics Co. Ltd. (“Samsung”) alleging the same on July 6, 2011. [Civil Action No.: 1:11-cv-4574 at D.I. 1] With respect to the ’750 patent, Cascades alleged that Samsung devices using Google Inc.’s Android operating system version 2.2 or higher of infringing claims 1 and 15 of the ’750 patent, and HTC devices using Google Inc.’s Android operating system version 2.2 or higher of infringing claim 15 of the ’750 patent.

After a consolidated claim construction proceeding for the Cascades cases against Samsung and HTC, the Court ordered separate trials, with Samsung’s trial commencing on July 13, 2015, and HTC’s trial commencing on September 14, 2015. [D.I. 178, Civil Action No.: 1:11-cv-4574 at D.I. 228]

Steven P. Mandell (ARDC #6183729)

Steven L. Baron (ARDC #6200868)

Mandell Menkes LLC

One North Franklin, Suite 3600

Chicago, Illinois 60606

Tel.: 312-251-1000

Fax: 312-251-1010

Attorneys for Defendant,

HTC Corporation

Exhibit D

IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF ILLINOIS
EASTERN DIVISION

CASCADES COMPUTER
INNOVATION, LLC.,

Plaintiff,

v.

MOTOROLA MOBILITY, INC.
and SAMSUNG ELECTRONICS CO.,
LTD.

Defendants.

Civil Action No. 1:11-cv-4574

Honorable Robert W. Gettleman

FILED
2/16/2012
THOMAS G. BRUTON
CLERK, U.S. DISTRICT COURT

THIRD AMENDED COMPLAINT FOR PATENT INFRINGEMENT

Cascades Computer Innovation, LLC ("Cascades") complains of Defendants Motorola Mobility, Inc. and Samsung Electronics Co., Ltd., and alleges the following:

NATURE OF THE SUIT

1. This is a suit for patent infringement arising under the patent laws of the United States, Title 35 of the United States Code § 1 et seq. This Court has exclusive jurisdiction over the subject matter of this Complaint under 28 U.S.C. § 1338(a).

PARTIES

2. Cascades is an Illinois limited liability company having its principal place of business at 500 Skokie Boulevard, Suite 350, Northbrook, IL 60062.

3. Defendant Motorola Mobility, Inc. ("Motorola") is a Delaware corporation headquartered at 600 N. U.S. Highway 45, Libertyville, Illinois 60048.

4. Samsung Electronics Co., Ltd. ("Samsung") is a foreign corporation with corporate headquarters at 250, 2-ga, Taepyung-ro, Jung-gu, Seoul 100-742, Republic of Korea.

5. Cascades' claim for patent infringement against Motorola and Samsung arises under the patent laws of the United States, including 35 U.S.C. §§271 and 281. Consequently, this Court has original subject matter jurisdiction over this suit pursuant to 28 U.S.C. §§1331 and 1338.

6. Motorola sells products throughout the United States and conducts substantial business in this judicial district, including providing the products accused of infringement in this judicial district. Motorola maintains its headquarters in this judicial district and also maintains retail sales offices in this judicial district.

7. Motorola is subject to both specific and general personal jurisdiction of this Court because, among other things, it has established continuous and systematic contacts with Illinois and in this judicial district; it has committed acts within Illinois and this judicial district giving rise to this action, and it has minimum contacts with the forum such that the exercise of jurisdiction over it would not offend traditional notions of fair play and substantial justice. As an example, Motorola has established distribution networks placing products that are covered by claims of Cascades' United States Patent No. 7,065,750 into the stream of commerce such that those products flow into Illinois and this district. Motorola has also committed acts of patent infringement and/or contributed to others' acts of patent infringement within this district.

8. Samsung sells products throughout the United States and conducts substantial business in this judicial district, including providing the products accused of infringement in this judicial district.

9. Samsung is subject to both specific and general personal jurisdiction of this Court because, among other things, it has established continuous and systematic contacts with Illinois and in this judicial district; it has committed acts within Illinois and this judicial district giving rise to this action, and it has minimum contacts with the forum such that the exercise of jurisdiction over it would not offend traditional notions of fair play and substantial justice. As an example, Samsung has established distribution networks placing products that are covered by claims of Cascades' United States Patent No. 7,065,750 into the stream of commerce such that those products flow into Illinois and this district. Samsung has also committed acts of patent infringement and/or contributed to others' acts of patent infringement within this district.

10. Venue is proper in this judicial district under 28 U.S.C. §§ 1391 and/or 1400(b).

PATENT AT ISSUE

11. On June 20, 2006, United States Patent No. 7,065,750 ("the '750 patent"), entitled "Method and Apparatus for Preserving Precise Exceptions in Binary Translated Code," was duly and legally issued by the United States Patent and Trademark Office. Cascades owns the exclusive license and right to sue for past, present and future infringement of the '750 Patent.

12. Motorola is now and has been infringing and/or contributorily infringing the '750 patent, literally and under the doctrine of equivalents, by, among other things, using, offering to sell, selling, re-selling and/or importing products that are covered by one or more claims of the '750 patent. Such infringing products include, but are not limited to, cell phone products such as its Motorola DROID³ smartphones. An illustrative claim chart demonstrating how Motorola and its customers practice the

method of claim 15 of the '750 patent using the Motorola DROID smartphones with their Android operating system and software is attached as Exhibit A. Other Motorola products that use the Android operating system and software infringe in the same way.

In fact, Motorola openly admits this on its website:

BASKING RIDGE, N.J., and LIBERTYVILLE, Ill. – July 7, 2011 – Verizon Wireless and Motorola Mobility, Inc. (NYSE: MMI), today announced the new Android™-powered DROID 3 by Motorola, a global smartphone that delivers power for work and play without making compromises.

DROID 3 by Motorola is the world's thinnest full QWERTY smartphone, and still delivers the power of a dual-core 1 GHz processor for fast multi-tasking. Customers can take stunning photos with the 8-megapixel camera or capture the moment in 1080p HD video. Equipped with Android 2.3, the DROID 3 by Motorola features a brilliant 4-inch qHD display, a roomy 5-row QWERTY keyboard and 3G Mobile Hotspot capabilities, with the ability to connect up to five Wi-Fi-enabled devices. DROID 3 by Motorola delivers the power needed to conquer the day whether customers are at home, work or somewhere in between.

Additional features

- Powered by Android™ 2.3 Gingerbread

[http://mediacenter.motorola.com/Press-Releases/Verizon-Wireless-Introduces-the-Next-Generation-Droid-Delivering-Power-and-Performance-The-Droid-3-by-Motorola-](http://mediacenter.motorola.com/Press-Releases/Verizon-Wireless-Introduces-the-Next-Generation-Droid-Delivering-Power-and-Performance-The-Droid-3-by-Motorola-3735.aspx)

[3735.aspx](http://mediacenter.motorola.com/Press-Releases/Verizon-Wireless-Introduces-the-Next-Generation-Droid-Delivering-Power-and-Performance-The-Droid-3-by-Motorola-3735.aspx). Motorola's personnel likewise demonstrate Motorola's Droid smartphones using Android (See, e.g., <http://www.youtube.com/watch?v=gJyRGc7juG4>), and infringement occurs through this use in the manner described in Exhibit A. Most recently, Motorola's Senior Vice President, Alain Mutricy, demonstrated Motorola's Droid 4, the successor to the Droid 3, which he indicates is being upgraded to Android 4.0: <http://www.youtube.com/watch?v=GHNZlpwBx4o>

13. Further, Motorola is a contributory infringer of the '750 patent because it knew at least when the original complaint was filed and, thus, knew then (and now

knows) that the steps of the claimed method described in claim 15, for example, were carried out by users of Motorola's phones, such as the Motorola DROID³ smartphone, that employ Android operating systems and software. Direct infringement by customer/end-users is illustrated in the attached Exhibit A. Motorola's customers are direct infringers and Motorola contributes to their infringement in that Motorola's customers carry out each and every step of the method defined, for example, in claim 15, as illustrated in the claim chart, Exhibit A. Motorola is fully aware of such direct infringement and encourages, aids, and assists it. Motorola also knows there are no substantial non-infringing uses of the accused products and Motorola knows its phones using the Android operating system and software are especially designed and made to use the Android operating system and software and, thus, are designed to practice, for example, the method of claim 15 of the '750 patent. Motorola knows this occurs when a Motorola customer/end-user operates a phone in the manner Motorola directs, instructs and teaches.

14. Motorola does more than simply sell products that use the Android operating system. It directs customers to use and shows them how to use the Android operating system and, thus, facilitates such use. Motorola's customers, in turn, actually use the identified cell phones to carry out each step of method claim 15, which is an act of direct infringement. The conditions of such use are known to Motorola and set forth in Exhibit A. Specifically, use of the method occurs when the phone is turned on and made functional. When the Android operating system starts up, the Dalvik Virtual Machine in the phone looks through the applications installed on the phone and builds a tree of dependencies. This dependency tree optimizes the byte code for every application and stores it in the Dalvik cache. The applications are then run using the

optimized byte code. Motorola is fully aware that its phones using Android operating systems are designed to operate and do, in fact, operate in this way. The claimed method is performed when a user of the cellular phones operates the device for their intended purpose using the Android operating system, e.g., allowing the Dalvik Virtual Machine to optimize the byte code for each application.

15. Further, Motorola's manufacture, sale, offer to sell and importation of the identified cell phones is a direct infringement of apparatus/system claim 1, for example, in that smartphones made and sold by Motorola such as the DROID³ smartphone include each element of claim 1, including a non-optimizing foreign code execution module, an optimizing binary translator, a host CPU, a documentation tracker and a recovery mechanism, all as required by claim 1.

16. Motorola makes the DROID³ cellular telephone.

17. Motorola DROID³ phone includes both hardware and software components.

18. Motorola imports, sells, and offers to sell the DROID³ phone to customers in the United States.

19. Motorola customers use the DROID³ phones in the United States.

20. The Motorola DROID³ phone:

- a. includes a binary translation system.
- b. includes an optimizing translator.
- c. includes a binary translated code translated from foreign code.
- d. designates a set of recovery points in the optimized binary translated code during optimized translation of the foreign code.

- e. generates a set of documentations during the optimized translation of the foreign code.

21. The Motorola DROID³ phone uses one of the documentations in the set of documentations corresponding to executed optimized binary translated code when an exception arises during its execution to recover a foreign state corresponding to a recovery point for the exception.

22. The Motorola DROID³:

- a. includes a non-optimizing foreign code execution module.
- b. includes a non-optimizing foreign code execution module dedicated to foreign state for each binary operation executed.
- c. includes an optimizing binary translator.
- d. includes a binary translator configured to translate foreign binary operations into optimized sequences of host operations.
- e. includes a host CPU.
- f. includes a host CPU configured to execute host operations.
- g. includes a host CPU to execute the host operations.
- h. includes a documentation generator.
- i. includes a document generator configured to generate a set of documentations for optimized sequences of host operations.
- j. includes a documentation tracker.
- k. includes a documentation tracker configured to record host operation addresses at appointed points of the host operation sequences being executed.
- l. includes a recovery mechanism.

m. includes a recovery mechanism configured to select a documentation in the set of documentations using a host operation address corresponding to the selected documentation.

23. Samsung is now and has been infringing and/or contributorily infringing the '750 patent, literally and under the doctrine of equivalents, by, among other things, using, offering to sell, selling, re-selling and/or importing products that are covered by one or more claims of the '750 patent. Such infringing products include, but are not limited to, cell phone products such as its Samsung Nexus S smartphones. An illustrative claim chart demonstrating how Samsung practices the method of claim 15 of the '750 patent using the Samsung Nexus smartphones with their Android operating system and software is attached as Exhibit B. Other Samsung products that use the Android operating system and software infringe in the same way. In fact, Samsung openly admits this:

OVERLAND PARK, Kan. – March 21, 2011 – Sprint (NYSE: S) extends its 4G device innovation lead once again with the upcoming availability of the 20th 4G device and fourth 4G phone, **Nexus S™ 4G1** from Google™. Coming to Sprint this spring, it will also be able to take advantage of the unprecedented controls and services enabled by Google Voice™ integration built into the Sprint Network.. Manufactured by Samsung Telecommunications America (Samsung Mobile), a leading global mobile phone provider and the No. 1 mobile phone provider in the United States2, Nexus S 4G comes packed with a pure Google experience using Android™ 2.3, Gingerbread, the fastest version of Android available for smartphones. It is powered by a 1GHz Samsung application processor that produces rich 3D-like graphics, faster upload and download times and supports HD-like multimedia content along with a dedicated Graphics Processing Unit (GPU) to make playing mobile games, browsing the Web and watching videos a fast, fluid and smooth experience.

“Nexus S 4G shows the strong commitment Sprint has to Android, and when combined with our 4G network capabilities, it gives customers the option of a pure Google experience,” said Fared Adib, vice president – Product Development, Sprint. “As the first 4G smartphone with Android 2.3, Nexus S 4G delivers on the promise of the advanced data capabilities

of 4G to deliver an incredible Web browsing experience, offers quick and easy access to future Android updates and access to the services built into Google Voice.”

It is designed with Samsung’s brilliant Super AMOLED™ touchscreen technology providing a premium viewing experience. The 4-inch Contour Display features a curved design for a more comfortable look and feel in the user’s hand or along the side of the face. It also offers a screen that is bright with higher color contrast, meaning colors are incredibly vibrant and text is crisp at any size and produces less glare than on other smartphone displays when outdoors, so videos, pictures and games look their best and the sun won’t wash them out

Sprint Nexus S 4G customers will be among the first to receive Android software upgrades and new Google mobile apps. In many cases, the device will get the updates and new apps as soon as they are available.

“We’re excited to partner with Sprint on Nexus S 4G, which brings innovative hardware by Samsung and innovations on the Android platform, to create a powerful smartphone experience,” said Andy Rubin, vice president of Engineering at Google.

(http://www.samsung.com/us/news/newsRead.do?news_seq=19830&page=18&gltype=localnews.) Samsung employees and corporate officers, including J.K. Shin, the President and Head of Mobile Communications at Samsung Electronics, and Kevin Packingham, Senior Vice President of Product Innovations, demonstrated a successor to the Nexus S, the Galaxy Nexus smartphone at a recent keynote also broadcast via webcam to introduce this latest model of Samsung phone with the Android OS.

(<http://androidandme.com/2011/10/news/video-miss-the-google-and-samsung-event-last-night-catch-the-full-keynote/>)

24. Further, Samsung is a contributory infringer of the ‘750 patent because it knew at least when the original complaint was filed and, thus, knew then (and now knows) that the steps of the claimed method described in claim 15, for example, were carried out by users of Samsung’s phones, such as the Samsung Nexus S smartphone, employing Android operating systems and software. Direct infringement by

customer/end-users is illustrated in the attached Exhibit B. Samsung's customers are direct infringers and Samsung contributes to their infringement in that Samsung's customers carry out each and every step of the method defined, for example, in claim 15, as illustrated in the claim chart, Exhibit B. Samsung is fully aware of such direct infringement and encourages, aids and assists it. Samsung also knows there are no substantial non-infringing uses of the accused products and Samsung knows its phones using the Android operating system and software are especially designed and made to use the Android operating system and software and, thus, are designed to practice, for example, the method of claim 15 of the '750 patent. Samsung knows this occurs when a Samsung customer/end-user operates a phone in the manner Samsung directs, instructs and teaches.

25. Samsung does more than simply sell products that use the Android operating system. It directs customers to use and shows them how to use the Android operating system and, thus, facilitates such use. Samsung's customers, in turn, actually use the identified cell phones to carry out each step of method claim 15, which is an act of direct infringement. The conditions of such use are known to Samsung and set forth in Exhibit B. Specifically, use of the method occurs when the phone is turned on and made functional. When the Android operating system starts up, the Dalvik Virtual Machine in the phone looks through the applications installed on the phone and builds a tree of dependencies. This dependency tree optimizes the byte code for every application and stores it in the Dalvik cache. The applications are then run using the optimized byte code. Samsung is fully aware that its phones using Android operating systems are designed to operate and do, in fact, operate in this way. The claimed method is performed when a user of the cellular phones operates the device for their

intended purpose using the Android operating system, e.g., allowing the Dalvik Virtual Machine to optimize the byte code for each application.

26. Further, Samsung's manufacture, sale, offer to sell and importation of the identified cell phones is a direct infringement of apparatus/system claim 1, for example, in that smartphones made and sold by Samsung such as the Nexus S smartphone include each element of claim 1, including a non-optimizing foreign code execution module, an optimizing binary translator, a host CPU, a documentation tracker and a recovery mechanism, all as required by claim 1.

27. Samsung makes the Nexus S cellular telephone.

28. Samsung's Nexus S phone includes both hardware and software components.

29. Samsung imports, sells, and offers to sell the Nexus S phone to customers in the United States.

30. Samsung customers use the Nexus S phones in the United States.

31. The Samsung Nexus S phone:

- a. includes a binary translation system.
- b. includes an optimizing translator.
- c. includes a binary translated code translated from foreign code.
- d. designates a set of recovery points in the optimized binary translated code during optimized translation of the foreign code.
- e. generates a set of documentations during the optimized translation of the foreign code.

32. The Samsung Nexus S phone uses one of the documentations in the set of documentations corresponding to executed optimized binary translated code when an

exception arises during its execution to recover a foreign state corresponding to a recovery point for the exception.

33. The Samsung Nexus S:

- a. includes a non-optimizing foreign code execution module.
- b. includes a non-optimizing foreign code execution module dedicated to foreign state for each binary operation executed.
- c. includes an optimizing binary translator.
- d. includes a binary translator configured to translate foreign binary operations into optimized sequences of host operations.
- e. includes a host CPU.
- f. includes a host CPU configured to execute host operations.
- g. includes a host CPU to execute the host operations.
- h. includes a documentation generator.
- i. includes a document generator configured to generate a set of documentations for optimized sequences of host operations.
- j. includes a documentation tracker.
- k. includes a documentation tracker configured to record host operation addresses at appointed points of the host operation sequences being executed.
- l. includes a recovery mechanism.
- m. includes a recovery mechanism configured to select a documentation in the set of documentations using a host operation address corresponding to the selected documentation.

PRAYER FOR RELIEF

WHEREFORE, Cascades prays for the following relief:

- A. A judgment finding Motorola and Samsung have each infringed and contributorily infringed the '750 patent;
- B. A judgment that the '750 patent is valid and enforceable;
- C. A permanent injunction enjoining Motorola and Samsung, their agents, officers, assigns and others acting in concert with them, from infringing, inducing infringement of, and/or contributing to infringement of the '750 patent;
- D. An award of damages adequate to compensate Cascades for the infringement of the '750 patent that has occurred;
- E. An award of pre-judgment interest and post-judgment interest on the damages awarded;
- F. A determination that this is an exceptional case and an award of Cascades' attorneys' fees pursuant to 35 U.S.C. § 285 and any other applicable statute or law, and an award to Cascades of its costs; and,
- G. Such other relief as the Court deems equitable under the circumstances.

JURY DEMAND

Plaintiff demands a trial by jury on all issues triable to a jury.

Raymond P. Niro (rniro@nshn.com)
Arthur A. Gasey (gasey@nshn.com)
Paul C. Gibbons (gibbons@nshn.com)
Anna B. Folgers (afolgers@nshn.com)
NIRO, HALLER & NIRO
181 W. Madison, Suite 4600
Chicago, IL 60602
(312) 236-0733
Fax: (312) 236-3137

Attorneys for Cascades Computer
Innovation, LLC

CERTIFICATE OF SERVICE

The undersigned hereby certifies that on February 10, 2012 the foregoing **THIRD AMENDED COMPLAINT FOR PATENT INFRINGEMENT** was filed electronically with the Clerk of the Court for the Northern District of Illinois using the Court's Electronic Case Filing System, which will send notification to the registered participants of the ECF System as listed in the Court's Notice of Electronic Filing.

I certify that all parties in this case are represented by counsel who are CM/ECF participants.

Attorneys for Cascades Computer Innovation,
LLC

**EXHIBIT A TO THIRD
AMENDED COMPLAINT
FOR PATENT INFRINGEMENT**

US 7,065,750 Claim 15 v. Motorola Droid Family

US 7,065,750 to Babaian



(12) **United States Patent**
Babaian et al.

(10) **Patent No.:** US 7,065,750 B2
(45) **Date of Patent:** Jun. 20, 2006

(54) **METHOD AND APPARATUS FOR PRESERVING PRE-EXECPTIONS IN BINARY TRANSLATED CODE**

(75) **Inventors:** Boris A. Babaian, Moscow (RU);
Andrew V. Yakushev, Sergiev-Podol (RU);
Sergey A. Rozikov, Moscow (RU);
Vladimir M. Gochelin, Moscow (RU)

(73) **Assignee:** Elbrus International, Georgia, Iowa (US)

(*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 723 days.

(21) **Appl. No.:** 09/038,452
(22) **Filed:** Apr. 18, 2001

(65) **Prior Publication Data:**
US 2002/0092002 A1 Jul. 11, 2002

Related U.S. Application Data
(63) Continuation-in-part of application No. 09/505,652, filed on Feb. 17, 2000

(60) Provisional application No. 60/120,348, filed on Feb. 17, 1999; provisional application No. 60/120,376, filed on Feb. 17, 1999; provisional application No. 60/120,380, filed on Feb. 17, 1999; provisional application No. 60/120,457, filed on Feb. 17, 1999; provisional application No. 60/120,458, filed on Feb. 17, 1999; provisional application No. 60/120,459, filed on Feb. 17, 1999; provisional application No. 60/120,504, filed on Feb. 17, 1999.

(51) **Int. Cl.**
G06F 9/455 (2006.01)
(52) **U.S. Cl.** 717/136

(58) **Field of Classification Search:**
717/130, 136, 151, 186, 712/23, 24, 203, 712/12, 228, 230, 235, 244, 237, 714/15, 714/17, 20, 35, 48, 49, 702, 707, 747
See application file for complete search history.

(56) **References Cited**
U.S. PATENT DOCUMENTS
5,537,550 A * * 1996 Kuo et al. 712/24
(Continued)

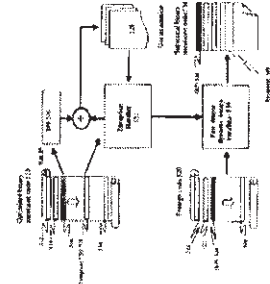
OTHER PUBLICATIONS
Chernoff, A., Hendry, M., Haskewy, R., Rovee, C., Rubin, N., Lee, J., Bhattacharjee, S., Yoon, J., "X12: a portable, distributed binary translator", Mar. Apr. 1998, IEEE Micro, p. 51-64, retrieved from IEEE Jul. 7, 2004.*

(Continued)
Primary Examiner: Wei Y. Zhou
Assistant Examiner: Mary Stebbins
(54) *Attorney, Agent, or Firm:* Townsend and Townsend and Crew LLP

(57) **ABSTRACT**

Pre-execption handling in the optimized binary native code is achieved by translating execution in the non-optimized source-code foreign code execution items in accordance with one of the several coherent foreign states designated during the optimized translation of the foreign code. A method to improve the operation by avoiding complete foreign state updates in the optimized code, an apparatus to track the switching between the states and a method to recompile the complete foreign state in accordance to the current state identification, execution context and additional documentation provided during the translation time are proposed.

18 Claims, 4 Drawing Sheets



US 7,065,750 Claim 15

15. A method of recomputing a dedicated foreign state (A) in a binary translation system (B) from documentation generated by an optimizing translator (C) in a case of an exception arising during execution of optimized binary translated code translated from a foreign code (D), the method comprising: designating a set of recovery points in the optimized binary translated code during optimized translation of the foreign code, wherein each recovery point represents a foreign state (E), generating a set of documentations during the optimized translation of the foreign code (F), wherein each documentation in the set of documentations corresponds to a recovery point in the optimized binary translated code and describes operations required to calculate a corresponding foreign state for the recovery point (G), and using one of the documentations in the set of documentations corresponding to executed optimized binary translated code when an exception arises during its execution to recover a foreign state corresponding to a recovery point for the exception, wherein the foreign state is recovered by executing the operations for the one of the documentations (H).

US 7,065,750 Claim 15

foreign state (A)

```
/* Create the PC reconstruction slot if not already done */
extern ArmLIR *genCheckCommon(CompilationUnit *cUnit, int dOffset,
                               ArmLIR *branch,
                               ArmLIR *pcLabel)
{
    /* Forget all def info (because we might rollback here. Bug #1367397 */
    dvmCompilerResetDefTracking(cUnit);

    /* Set up the place holder to reconstruct this Dalvik PC */
    if (pcLabel == NULL) {
        int dPC = (int) (cUnit->method->insns + dOffset);
        pcLabel = dvmCompilerNew(sizeof(ArmLIR), true);
        pcLabel->opCode = kArmPseudoPCReconstructionCell;
        pcLabel->operands[0] = dPC;
        pcLabel->operands[1] = dOffset;
        /* Insert the place holder to the growable list */
        dvmInsertGrowableList(&cUnit->pcReconstructionList, pcLabel);
    }

    /* Branch to the PC reconstruction code */
    branch->generic.target = (LIR *) pcLabel;
    return pcLabel;
};
```

Note: Dalvik PC – Program Counter for the interpreted (foreign) code

Dalvik code analysis, CodegenCommon.c

US 7,065,750 Claim 15

a binary translation system

Note: JIT is part of all the releases of Android since 2.2. Source – code analysis

Dalvik JIT

Posted by Tim Bray on 25 May 2010 at 2:57 PM

[This post is by Dan Bornstein, virtual-machine wrangler. — Tim Bray]

As the tech lead for the Dalvik team within the Android project, I spend my time working on the virtual machine (VM) and core class libraries that sit beneath the Android application framework. This layer is mostly invisible to end users, but done right, it helps make Android devices run smoothly and improves developer productivity.

The 2.2 release is particularly pleasing to me, as it is the first release since before 1.0 in which we have been able to deliver significantly new VM technology. And unlike much of what my team and I do, it is something that can be experienced directly by end users.

"Dalvik" isn't exactly a household word (at least in my country), and most people wouldn't know a virtual machine if it hit them in the face, but when you tell them you were able to make their existing device work better — run faster, use less battery — they will actually take notice!

What Makes This Possible?

We added a Just In Time (JIT) compiler to the Dalvik VM. The JIT is a software component which takes application code, analyzes it, and actively translates it into a form that runs faster, doing so while the application continues to run. If you want to learn more about the design of the Dalvik JIT, please watch the excellent talk from Google I/O 2010 given by my colleagues Bill Buzbee and Ben Cheng, which should be posted to YouTube very soon.



From: Android developers blog, <http://android-developers.blogspot.com/2010/05/dalvik-jit.html>

US 7,065,750 Claim 15

a binary translation system

DROID BY MOTOROLA



ANDROID™ PLATFORM
Android 2.3 (Gingerbread)

Source: <http://www.motorola.com/Consumers/US-EN/Consumer-Product-and-Services/Mobile-Phones/ci.DROID-3-by-MOTOROLA-US-EN.alt>

US 7,065,750 Claim 15

optimizing translator (G)

Dalvik Interpreter

- Dalvik programs consist of byte code, processed by a host-specific interpreter
 - Highly-tuned, very fast interpreter (2x similar)
 - Typically less than 1/3rd of time spent in the interpreter
 - OS and performance-critical library code natively compiled
 - Good enough for most applications
- Performance a problem for compute-intensive applications
 - Partial solution was the release of the Android Native Development Kit, which allows Dalvik applications to call out to statically-compiled methods
- Other part of the solution is a Just-In-Time Compiler
 - Translates byte code to optimized native code at run time



Source: <http://dl.google.com/googleio/2010/android-jit-compiler-androids-dalvik-vm.pdf>

US 7,065,750 Claim 15

binary translated code
translated from a foreign
code (D),

Dalvik Interpreter

- Dalvik programs consist of byte code, processed by a host-specific interpreter
 - Highly-tuned, very fast interpreter (2x similar)
 - Typically less than 1/3rd of time spent in the interpreter
 - OS and performance-critical library code natively compiled
 - Good enough for most applications
- Performance a problem for compute-intensive applications
 - Partial solution was the release of the Android Native Development Kit, which allows Dalvik applications to call out to statically-compiled methods
- Other part of the solution is a Just-In-Time Compiler
 - Translates byte code to optimized native code at run time



Source: <http://dl.google.com/googleio/2010/android-jit-compiler-androids-dalvik-vm.pdf>

US 7,065,750 Claim 15

designating a set of recovery points in the optimized binary translated code during optimized translation of the foreign code, wherein each recovery point represents a foreign state

```
/* Create the PC reconstruction state if not already done */
extern ArmLIR *genCheckCommon(CompilationUnit *cUnit, int doffset,
                               ArmLIR *branch,
                               ArmLIR *pcLabel)
{
    /* Forget all def info (because we might rollback here. Bug #235739) */
    dvmCompilerResetDefTracking(cUnit);

    /* Set up the place holder to reconstruct this Dalvik PC */
    if (pcLabel == NULL) {
        int dPC = (int) (cUnit->method->insns + doffset);
        pcLabel = dvmCompilerNew(sizeof(ArmLIR), true);
        pcLabel->opCode = kArmPseudoPCReconstructionCell;
        pcLabel->operands[0] = dPC;
        pcLabel->operands[1] = doffset;
        /* Insert the place holder in the pcReconstructionList */
        dvmInsertGrowableViewList(&cUnit->pcReconstructionList, pcLabel);
    }

    /* Branch to the PC reconstruction code */
    branch->generic.target = (LIR *) pcLabel;
    return pcLabel;
}
```

Dalvik code analysis, CodegenCommon.c

US 7,065,750 Claim 15

generating a set of documentations during the optimized translation of the foreign code (F),

```
/* Read the Dalvik PC into r0 and jump to the specified target */
static void handlePCReconstruction(CompilationUnit *cUnit,
                                   ARMIR *targetLabel)
{
    ARMIR **pcLabel =
        (ARMIR **) cUnit->pcReconstructionList.elemList;
    int numElems = cUnit->pcReconstructionList.numUsed;
    int i;
    for (i = 0; i < numElems; i++) {
        dvmCompilerAppendIR(cUnit, (LIR *) pcLabel[i]);
        /* r0 = dalvik PC */
        loadConstant(cUnit, r0, pcLabel[i]->operands[0]);
        genUnconditionalBranch(cUnit, targetLabel);
    }
}
```

Dalvik code analysis, CodegenDriver.c

US 7,065,750 Claim 15

each documentation in the set of documentations corresponds to a recovery point in the optimized binary translated code and describes operations required to calculate a corresponding foreign state for the recovery point

```
/* Load the Dalvik PC into r0 and jump to the specified target */
static void handlePCReconstruction(CompilationUnit *cUnit,
                                   ArmLIR *targetLabel)
{
    ArmLIR *pcrLabel =
        (ArmLIR *) cUnit->pcReconstructionList.elemList;
    int numElems = cUnit->pcReconstructionList.numUsed;
    int i;
    for (i = 0; i < numElems; i++) {
        dvmCompilerAppendLIR(cUnit, (LIR *) pcrLabel[i]);
        /* r0 = Dalvik PC */
        loadConstant(cUnit, r0, pcrLabel[i]->operands[0]);
        genUnconditionalBranch(cUnit, targetLabel);
    }
}
```

Dalvik code analysis, CodegenDriver.c

US 7,065,750 Claim 15

using one of the documentations in the set of documentations
corresponding to executed optimized binary translated code when an
exception arises during its execution to recover a foreign state
corresponding to a recovery point for the exception, wherein the foreign
state is recovered by executing the operations for the one of the
documentations (44).


```
/* Now create a special block to hint PC reconstruction */
lastBB->next = dvmCompilerNewBB(kPCReconstruction);
lastBB = lastBB->next;
lastBB->id = numBlocks++;

/* And one final block that publishes the PC and raise the exception */
lastBB->next = dvmCompilerNewBB(kExceptionHandler);
lastBB = lastBB->next;
lastBB->id = numBlocks++;
```

Continued

Dalvik code analysis, FrontEnd.c

US 7,065,750 Claim 15

using one of the documentations in the set of documentations
corresponding to executed optimized binary translated code when an
exception arises during its execution to recover a foreign state
corresponding to a recovery point for the exception, wherein the foreign
state is recovered by executing the operations for the one of the
documentations .

```
void dvmCompilerMIR2LIR(CompilationUnit *cUnit)
{
    .....
```

```
    case kExceptionHandler:
        labelList[i].opCode = kAmpPseudoEHBlockLabel;
        if (cUnit->pcReconstructionList.numUsed) {
            loadWordDisp(cUnit, xGLUE, offsetof(interpState,
                jitToInterpEntries.dvmJitToInterpFun),
                r1);
            opReg(cUnit, kOpBlx, r1);
        }
        break;
```

```
* 3) dvmJitToInterpFun: use the fast interpreter to execute the next
* instruction(s) and stay there as long as it is appropriate to return
* to the compiled land. This is used when the jit'ed code is about to
* throw an exception.
```

Continued

Dalvik code analysis, codegendriver.c, InterpDefs.h

US 7,065,750 Claim 15

using one of the documentations in the set of documentations
corresponding to executed optimized binary translated code when an
exception arises during its execution to recover a foreign state
corresponding to a recovery point for the exception, wherein the foreign
state is recovered by executing the operations for the one of the
documentations .

```
.global dvmJitToInterpPunt
dvmJitToInterpPunt:
    ldr    r10, [xGLUE, #offset_glue_self]    @ callee saved r10 <- glue->self
    mov    r2, #kSVSPunt                     @ r2<- interpreter entry point
    mov    r3, #0
    str    r3, [r10, #offset_initCodeCache] @ Back to the interp land
    b      jitSVShadowRunEnd                  @ doesn't return
```

Dalvik code analysis, footer.S

EXHIBIT B TO THIRD AMENDED COMPLAINT FOR PATENT INFRINGEMENT

US 7,065,750 Claim 15 v. Nexus/Galaxy Family

US 7,065,750 to Babaian



(12) **United States Patent**
Babaian et al.

(10) **Patent No.:** **US 7,065,750 B2**
(45) **Date of Patent:** **Jun. 20, 2006**

(54) **METHOD AND APPARATUS FOR
PRESERVING PRECISE EXCEPTIONS IN
BINARY TRANSLATED CODE**

(57a) **Inventors:** **Borib A. Babaian, Moscow (RU);
Andrew V. Vokoshev, Sergey-Polud
(RU); Sergey A. Rozikov, Moscow
(RU); Vladimir M. Gashulin, Moscow
(RU)**

(73) **Assignee:** **Ethern International, George Town
(KY)**

(*) **Notice:** Subject to any disclaimer, the term of this
patent is extended or adjusted under 35
U.S.C. 154(b) by 723 days.

(21) **Appl. No.:** **09/838,552**

(22) **Filed:** **Apr. 18, 2001**

(65) **Priority Publication Data**
US 2002/0092002 A1 Jul. 11, 2002

Related U.S. Application Data

(63) **Continuation-in-part of application No. 09/595,652,
filed on Feb. 17, 2000**

(60) **Provisional application No. 60/120,438, filed on Feb.
17, 1999; provisional application No. 60/120,376,
filed on Feb. 17, 1999; provisional application No.
60/120,380, filed on Feb. 17, 1999; provisional appli-
cation No. 60/120,457, filed on Feb. 17, 1999; pro-
visional application No. 60/120,438, filed on Feb. 17,
1999; provisional application No. 60/120,459, filed
on Feb. 17, 1999; provisional application No. 60/120,
504, filed on Feb. 17, 1999.**

(51) **Int. Cl.**

G06F 9/453

(2006.01)

(52) **U.S. Cl.** **717/136**

(58) **Field of Classification Search**
717/136, 137, 138, 139, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000.

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS
5,537,559 A * 1996. Kim et al. 712,244
(continued)

OTHER PUBLICATIONS

Chernoff, A., Hockley, M., Hockley, R., Rovee, C., Rubin, N.,
Tye, J., Bhargava, V., S. Yates, J., "X/M2: a profile-directed
binary translator", Mar. Apr. 1998, IBM Micro, p. 5464,
retrieved from IEEE Jul. 7, 2004.*

(continued)

Primary Examiner: **Wei Y. Chen**

Assistant Examiner: **Mary Sheehan**

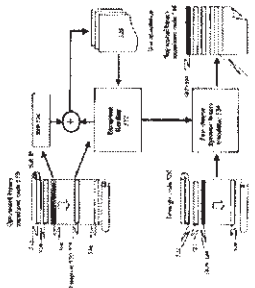
(74) **Attorney, Agent, or Firm:** **Townsend and Townsend**

(continued)

Abstract

Precise exceptions handling in the optimized binary trans-
lated code is achieved by translating exceptions in the
non-optimized assembly-code foreign code execution means in
accordance with one of the several coherent foreign states
designated during the optimized translation of the foreign
code. A method to improve the operation by avoiding
complete foreign state updates in the optimized code, an
apparatus to track the switching between the states and a
method to accomplish the complete foreign state in accor-
dance to the current state identification, execution context
and additional discussion provided during the transla-
tion time are proposed.

18 Claims, 4 Drawing Sheets



US 7,065,750 Claim 15

15. A method of recomputing a dedicated foreign state (A) in a binary translation system (B) from documentation generated by an optimizing translator (C) in a case of an exception arising during execution of optimized binary translated code translated from a foreign code (D), the method comprising: designating a set of recovery points in the optimized binary translated code during optimized translation of the foreign code, wherein each recovery point represents a foreign state (E); generating a set of documentations during the optimized translation of the foreign code (F), wherein each documentation in the set of documentations corresponds to a recovery point in the optimized binary translated code and describes operations required to calculate a corresponding foreign state for the recovery point (G); and using one of the documentations in the set of documentations corresponding to executed optimized binary translated code when an exception arises during its execution to recover a foreign state corresponding to a recovery point for the exception, wherein the foreign state is recovered by executing the operations for the one of the documentations (H).

US 7,065,750 Claim 15

foreign state (A)

```
/* Create the PC reconstruction slot if not already done */
extern ArmLIR *genCheckCommon(CompilationUnit *cUnit, int dOffset,
                              ArmLIR *branch,
                              ArmLIR *pcLabel)
{
    /* Forget all def info (because we might rollback here. Bug #236739) */
    dvmCompilerResetDefTracking(cUnit);

    /* Set up the place holder to reconstruct the Dalvik PC */
    if (pcLabel == NULL) {
        int dPC = (int) (cUnit->method->insns + dOffset);
        pcLabel = dvmCompilerNew(sizeof(ArmLIR), true);
        pcLabel->opCode = kArmPseudoPCReconstructionCell;
        pcLabel->operands[0] = dPC;
        pcLabel->operands[1] = dOffset;
        /* Insert the place holder to the growable list */
        dvmInsertGrowableList(&cUnit->pcReconstructionList, pcLabel);
    }

    /* Branch to the PC reconstruction code */
    branch->generic.target = (LIR *) pcLabel;
    return pcLabel;
}
```

Note: Dalvik PC – Program Counter for the interpreted (foreign) code

Dalvik code analysis, CodegenCommon.c

US 7,065,750 Claim 15

a binary translation system

Note: JIT is part of all the releases of Android since 2.2. Source – code analysis

Dalvik JIT

Posted by Tim Bray on 25 May 2010 at 2:57 PM

[This post is by Dan Bornstein, virtual-machine wrangler. — Tim Bray]

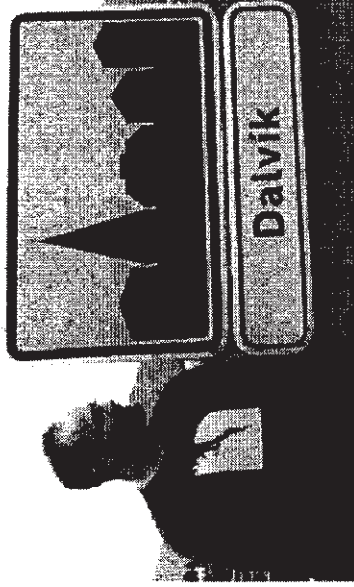
As the tech lead for the Dalvik team within the Android project, I spend my time working on the virtual machine (VM) and core class libraries that sit beneath the Android application framework. This layer is mostly invisible to end users, but done right, it helps make Android devices run smoothly and improves developer productivity.

The 2.2 release is particularly pleasing to me, as it is the first release since before 1.0 in which we have been able to deliver significantly new VM technology. And unlike much of what my team and I do, it is something that can be experienced directly by end users.

“Dalvik” isn’t exactly a household word (at least in my country), and most people wouldn’t know a virtual machine if it hit them in the face, but when you tell them you were able to make their existing device work better — run faster, use less battery — they will actually take notice!

What Makes This Possible?

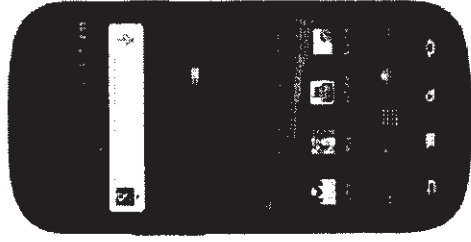
We added a Just In Time (JIT) compiler to the Dalvik VM. The JIT is a software component which takes application code, analyzes it, and actively translates it into a form that runs faster, doing so while the application continues to run. If you want to learn more about the design of the Dalvik JIT, please watch the excellent talk from Google I/O 2010 given by my colleagues Bill Buzbee and Ben Cheng, which should be posted to YouTube very soon.



From: Android developers blog, <http://android-developers.blogspot.com/2010/05/dalvik-jit.html>

US 7,065,750 Claim 15

a binary translation system



Meet the Nexus S with Android 2.3

Published: December 6, 2010

Samsung and Google deliver Nexus S, the world's first handset to feature the latest version of Google's Android™ platform. Powered by Android 2.3, Gingerbread, this smartphone is packed with powerful technology and the latest in hardware features.

Source: <http://www.samsung.com/us/article/meet-the-nexus-s-with-android-2-3>

US 7,065,750 Claim 15

optimizing translator (G)

Dalvik Interpreter

- Dalvik programs consist of byte code, processed by a host-specific interpreter
 - Highly-tuned, very fast interpreter (2x similar)
 - Typically less than 1/3rd of time spent in the interpreter
 - OS and performance-critical library code natively compiled
 - Good enough for most applications
- Performance a problem for compute-intensive applications
 - Partial solution was the release of the Android Native Development Kit, which allows Dalvik applications to call out to statically-compiled methods
- Other part of the solution is a Just-In-Time Compiler
 - Translates byte code to optimized native code at run time



Source: <http://dl.google.com/googleio/2010/android-jit-compiler-androids-dalvik-vm.pdf>

US 7,065,750 Claim 15

binary translated code
translated from a foreign
code (D),

Dalvik Interpreter

- Dalvik programs consist of byte code, processed by a host-specific interpreter
 - Highly-tuned, very fast interpreter (2x similar)
 - Typically less than 1/3rd of time spent in the interpreter
 - OS and performance-critical library code natively compiled
 - Good enough for most applications
- Performance a problem for compute-intensive applications
 - Partial solution was the release of the Android Native Development Kit, which allows Dalvik applications to call out to statically-compiled methods
- Other part of the solution is a Just-In-Time Compiler
 - Translates byte code to optimized native code at run time



Source: <http://dl.google.com/googleio/2010/android-jit-compiler-androids-dalvik-vm.pdf>

US 7,065,750 Claim 15

designating a set of recovery points in the optimized binary translated code during optimized translation of the foreign code, wherein each recovery point represents a foreign state (E)

```
/* Create the PC reconstruction list if not already done */
extern ArMLIR *genCheckCommon(CompilationUnit *cUnit, int doffset,
                               ArMLIR *branch,
                               ArMLIR *pcLabel)
{
    /* Forget all def info because we might rollback here. Bug #235739 */
    dvmCompilerResetDefTracking(cUnit);

    /* Set up the place holder to reconstruction table Dalvik PC */
    if (pcLabel == NULL) {
        int dPC = (int) (cUnit->method->insns + doffset);
        pcLabel = dvmCompilerNew(sizeof(ArMLIR), true);
        pcLabel->opCode = kArmPseudoPCReconstructionCell;
        pcLabel->operands[0] = dPC;
        pcLabel->operands[1] = doffset;
        /* Insert the place holder to the growable list */
        dvmInsertGrowableList(&cUnit->pcReconstructionList, pcLabel);
    }
    /* Branch to the PC reconstruction code */
    branch->generic.target = (LIR *) pcLabel;
    return pcLabel;
}
```

Dalvik code analysis, CodegenCommon.c

US 7,065,750 Claim 15

generating a set of documentations during the optimized translation of the foreign code (F),

```
/* Load the Dalvik PC into r0 and jump to the specified target */
static void handlePCReconstruction(CompilationUnit *cUnit,
                                   ArmLIR *targetLabel)
{
    ArmLIR **pcrLabel =
        (ArmLIR **) cUnit->pcReconstructionList.elemList;
    int numElems = cUnit->pcReconstructionList.numUsed;
    int i;
    for (i = 0; i < numElems; i++) {
        dvmCompilerAppendLIR(cUnit, (LIR *) pcrLabel[i]);
        /* r0 = Dalvik PC */
        loadConstant(cUnit, r0, pcrLabel[i]->operands[0]);
        genUnconditionalBranch(cUnit, targetLabel);
    }
}
```

Dalvik code analysis, CodegenDriver.c

US 7,065,750 Claim 15

each documentation in the set of documentations corresponds to a recovery point in the optimized binary translated code and describes operations required to calculate a corresponding foreign state for the recovery point

```
/* Replaced the Dalvik FC into r0 and jump to the specified target */
static void handlePCReconstruction(CompilationUnit *cUnit,
                                   ArmLIR *targetLabel)
{
    ArmLIR **pcrLabel =
        (ArmLIR **) cUnit->pcReconstructionList.elemList;
    int numElems = cUnit->pcReconstructionList.numUsed;
    int i;
    for (i = 0; i < numElems; i++) {
        dvmCompilerAppendLIR(cUnit, (LIR *) pcrLabel[i]);
        /* r0 = Dalvik FC */
        loadConstant(cUnit, r0, pcrLabel[i]->operands[0]);
        genUnconditionalBranch(cUnit, targetLabel);
    }
}
```

Dalvik code analysis, CodegenDriver.c

US 7,065,750 Claim 15

using one of the documentations in the set of documentations corresponding to executed optimized binary translated code when an exception arises during its execution to recover a foreign state corresponding to a recovery point for the exception, wherein the foreign state is recovered by executing the operations for the one of the documentations.

```
/* Now create a special block to host PC reconstruction code */
lastBB->next = dvmCompilerNewBB(kPCReconstruction);
lastBB = lastBB->next;
lastBB->id = numBlocks++;

/* And one final block that publishes the PC and raise the exception */
lastBB->next = dvmCompilerNewBB(kExceptionHandler);
lastBB = lastBB->next;
lastBB->id = numBlocks++;
```

Continued

Dalvik code analysis, FrontEnd.c

US 7,065,750 Claim 15

using one of the documentations in the set of documentations corresponding to executed optimized binary translated code when an exception arises during its execution to recover a foreign state corresponding to a recovery point for the exception, wherein the foreign state is recovered by executing the operations for the one of the documentations

```
void dvmCompilerMIR2LIR(CompilationUnit *cUnit)
{
    .....

```

```
        case kExceptionHandler:
            labelList[i].opCode = kAmpPseudoEhBlockLabel;
            if (cUnit->pcReconstructionList.numUsed) {
                loadWordDisp(cUnit, xGLUE, offsetof(interpState,
                    jitToInterpEntries.dvmJitToInterpPunt),
                    x1);
                opReg(cUnit, kOpBlx, x1);
            }
            break;

```

```

* 3) dvmJitToInterpPunt: use the fast interpreter to execute the next
* instruction(s) and stay there as long as it is appropriate to return
* to the compiled land. This is used when the jit'ed code is about to
* throw an exception.

```

Continued

Dalvik code analysis, codegendriver.c, InterpDefs.h

US 7,065,750 Claim 15

using one of the documentations in the set of documentations
corresponding to executed optimized binary translated code when an
exception arises during its execution to recover a foreign state
corresponding to a recovery point for the exception, wherein the foreign
state is recovered by executing the operations for the one of the
documentations .

```
.global dvmJitToInterpPunt
dvmJitToInterpPunt:
    ldr    r10, [rGLUE, #offset_self]    @ callee saved r10 <- glue->self
    mov    r2, #kSVSPunt                @ r2<- interpreter entry point
    mov    r3, #0
    str    r3, [r10, #offset_inJitCodeCache] @ Back to the interp land
    b      jitSVShadowRunEnd            @ doesn't return
```

Dalvik code analysis, footer.S

US 7,065,750 Applicability to other Samsung products

US 7,065,750 is also applicable to at least following Samsung products

Nexus S

Gem

Gem Touchscreen

Continuum

Mesmerize i500

Fascinate

Showcase i500

Showcase Galaxy S

Acclaim

Galaxy Indulge

Capacitive

Infuse 4G

Sidekick 4G

SG

Vibrant

Epic 4G

SG

Replenish

Galaxy Prevail

Intercept Galaxy Tab (tablet)

Galaxy S (tablet)

Exhibit E

IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF ILLINOIS
EASTERN DIVISION

CASCADES COMPUTER
INNOVATION, LLC.,

Plaintiff,

v.

MOTOROLA MOBILITY, INC.,
and SAMSUNG ELECTRONICS CO.,
LTD.,

Defendants.

Civil Action No. 11-cv-4574

Honorable Robert W. Gettleman

**CASCADES' OPPOSITION
TO MOTOROLA'S AND SAMSUNG'S MOTION TO DISMISS**

Defendants' latest dismiss motion continues their and their lawyers' efforts to delay this case. Contrary to Defendants' claims in moving yet again to dismiss, this case and the Sony-Ericsson case pending before Judge Darrah, are the only cases with defendants still pursuing a motion to dismiss Cascades' allegations of infringement of its '750 patent; the motions in both of these cases were filed by Motorola's lawyers. If one were to believe Motorola's and Samsung's latest motion, it is remarkable that HTC, LG, Dell, Pantech, Acer, and Sharp, along with their lawyers, are at all able to understand Cascades' claims of infringement and their applicability to the accused products – products akin to the Motorola and Samsung products accused of infringement in this case. Yet, all of these defendants have filed answers and in some cases counterclaims; certainly the ability of these other defendants to answer Cascades' claims of infringement of the '750 patent demonstrates that Cascades has sufficiently alleged infringement of claims 1 and 15 of the '750 patent, especially when Cascades' Third Amended Complaint in this case contains more detail than the pleadings Cascades filed in its other cases involving the '750 patent.

The fact that Motorola and Samsung have retreated to labeling Cascades' Third Amended Complaint as "misconduct" is revealing. A fair question is to ask what the motivation of these Defendants and their lawyers is in repeatedly engaging in motion practice that all other accused infringers have avoided. The answer is that these motions are designed to distract and delay, and to press for more detail that amounts to free, premature discovery for Defendants and additional burden for Cascades. Cascades' now Third Amended Complaint is not merely adequate or sufficient, but goes above and beyond the pleading requirements in accusing Motorola and Samsung of infringement of the '750 patent. Defendants' Motion to Dismiss should be denied.

I. SUMMARY OF THE THIRD AMENDED COMPLAINT

Cascades' Third Amended Complaint addresses all of the issues raised by Motorola and Samsung first in their joint motion to dismiss Cascades' Second Amended Complaint, and now by Defendants' pending dismiss motion. Cascades filed its Third Amended Complaint in order to end the issue of infringement that the Defendants and their lawyers seek to endlessly debate. Cascades' Third Amended Complaint (D.I. 36) satisfies the pleading requirements of the Federal Rules in that it:

- contains "a short and plain statement of the claim showing that the pleader is entitled to relief." Fed.R.Civ.P. 8(a)(2); and
- is written with "the simplicity and brevity that these [the Rules] contemplate." Fed.R.Civ.P. 84.

As they have all along, Motorola and Samsung continue to confuse the requirements of pleading (which is to give **notice** of the basis for a claim) with the requirements of a final judgment (which requires a party to satisfy its burden of proving every element of a claim on the merits). Edelson v. Ch'ien, 352 F. Supp. 2d 861, 870

(N.D. Ill. 2005); (“The purpose of a motion to dismiss pursuant to Rule 12(b)(6) is to test the legal sufficiency of a complaint, ***not the merits of the case***”) (emphasis added); Smith v. Med. Admin. Grp., Inc., 639 F.3d 277, 281 (7th Cir. 2011) (“[A plaintiff] need not plead a detailed set of facts, so long as the complaint supplies [defendant] with ‘fair notice of what ... the claim is and the grounds upon which it rests’” (quoting Erickson v. Pardus, 551 U.S. 89, 93 (2007))). Defendants’ attempt to denigrate the inventions of the ‘750 patent by equating them to “small fragments” or “few snippets” of “computer code” has no bearing upon whether Cascades has sufficiently pleaded infringement.

As Judge Darrah held in Edge Capture L.L.C. v. Lehman Bros. Holdings, Inc., 2008 U.S. Dist. LEXIS 83945 (N.D. Ill. Aug. 28, 2008):

The Rule 12(b)(6) pleading requirements for a complaint of infringement cannot be extended to require a plaintiff to specifically include each element of the claims of the asserted patent.” Phonometrics, Inc. v. Hospitality Franchise Systems, Inc., 203 F.3d 790, 794 (Fed. Cir. 2000). Rather, “a patentee need only plead facts sufficient to place the alleged infringer on notice.” Phonometrics, 203 F.3d at 794. See also McZeal v. Sprint Nextel Corp., 501 F.3d 1354, 1356-57 (Fed. Cir. 2007) (holding that a complaint that included the elements set out in the form complaint for patent infringement of the Federal Rules of Civil Procedure - (1) an allegation of jurisdiction; (2) a statement that the plaintiff owns the patent; (3) a statement that defendant has been infringing the patent “by making, selling, and using [the device] embodying the patent”; (4) a statement that the plaintiff has given the defendant notice of its infringement; and (5) a demand for an injunction and damages - satisfied the pleading requirements). The Complaint states that “[t]he technology and methods of automated trading [used by Defendants] infringe directly and/or indirectly one or more claims of the ‘833 patent and/or one or more claims of the ‘629 patent.” This statement sufficiently puts Defendants on notice. Further specificity is not required at this stage of the proceedings.

2008 U.S. Dist. LEXIS 83945 at *3-4. The Edge Capture decision applies with even greater force to the facts of the present case. In Edge Capture, the complaint at issue alleged simply that the defendants’ “use and/or direct the use of technology to perform automated/algorithmic trading of derivatives and/or underlying securities” infringes on one or more claims of the asserted patents. Id. at *2. In other words, in Edge Capture

no specific product was identified, much less an explanation as to how such conduct constituted infringement.

In briefing Sony-Ericsson's dismiss motion currently pending before Judge Darrah in the case styled Cascades v. Sony Ericsson, 1:11-cv-7223, Sony's lawyers, also representing Motorola in this case, devoted much of their argument to distinguishing Edge Capture and Cascades expects Motorola, as well as Samsung, to parrot those arguments in reply to Cascades' briefing in this case. Those arguments by Motorola's lawyers as to the supposed inapplicability of Edge Capture are misplaced.

Specifically, Motorola's lawyers argue that Edge Capture cannot apply to the present case where an infringer (such as Motorola) was not responsible for developing the software in the accused infringing products. This logic would lead to the perverse conclusion that independent developers of software in an infringing product would infringe, while those who take or even steal the same software from a third party could somehow escape infringement. In any event, this argument by Motorola's lawyers studiously ignores existing case law from Judge Kendall and followed by other decisions in this District, which holds that the requisite notice for a pleading to survive a motion to dismiss may be provided through the existence of the initial complaint in the case. That is, by virtue of Cascades' Complaint against Motorola and Samsung, those accused infringers had sufficient notice such that their continuing sales and support for their customers' use of the infringing devices is sufficient to support a claim of indirect infringement in a subsequent Amended Complaint. Trading Tech. Int'l, Inc. v. BCG Partners, Inc., 2011 U.S. Dist. LEXIS 99415, at *15-16 (N.D. Ill. Sept. 2, 2011).

A second alleged point of distinction for Edge Capture offered by Motorola's lawyers is the supposed ability of users to disable the software in question, or their

ability to use the accused device outside of the United States. (See Cascades v. Sony Ericsson, D.I. 27 at 15). Those arguments are speculative and simply inapposite to the sufficiency of Cascades' pleadings. It is no doubt possible that a user could conceivably erase all of the software on an accused phone, just as it is theoretically possible that a purchaser could simply destroy the phone without any use. What Cascades has pled, and what it has shown evidence of, is that real world end-users are using the accused Motorola and Samsung devices in their intended manner, e.g., with the Android software installed by Motorola and Samsung, and in the manner in which the use of the phone was intended. Edge Capture is applicable to the facts of the present case.

Nonetheless, in Edge Capture this Court found that such a pleading survived a motion to dismiss under Twombly. By comparison, Cascades' Amended Complaint (D.I. 23) in the present case identifies specific products, specific claims of the asserted patent, and even attaches a claim chart to lay out the facts supporting the elements of the infringement claims. In short, using this Court's rationale in Edge Capture, Cascades' Amended Complaint alone goes far beyond the specificity required at this point in the case, and Cascades' Third Amended Complaint (D.I. 36) goes still further.

Cascade's Third Amended Complaint not only provides notice of the claims made against Motorola and Samsung, it contains detail beyond anything required under the law. The basis for jurisdiction and venue are pled (D.I. 36, ¶¶ 5-10). The patent at issue is identified (Id., ¶ 11). The basis for direct infringement of method claim 15 is described -- Motorola and Samsung have themselves practiced the claimed method through public and private **use** of their smartphones running the Android OS, such as their DROID³ and Nexus S smartphones (Id., ¶ 12). Use of a claimed method by each Defendant and its customers is an act of direct infringement. BMC Res., Inc. v.

Paymentech, Lp, 498 F.3d 1373, 1378 (Fed. Cir. 2007); Centillion Data Sys. v. Qwest Communications, 631 F.3d 1279, 1284 (Fed. Cir. 2011).

Direct infringement is also pled for system/apparatus claim 1 (D.I. 36, ¶¶15 and 26). This second act of direct infringement takes place independently as a result of Motorola's and Samsung's sale, offer for sale, importation or use of the DROID³ or Nexus S smartphones, among others of Defendants' Android-based products. Id.

In paragraphs 13-14 and 24-25, all of the elements of contributory infringement are also pled with specificity: (1) that Motorola and Samsung knew of the '750 patent; (2) that direct infringement occurs by Motorola's and Samsung's customers; (3) that Motorola and Samsung know of their direct infringement, know that there are no substantial non-infringing uses, and know their products are especially made to infringe (Id., ¶ 13). The manner and method of infringement is also described in detail, as is how the method is used (Id., ¶ 14). Direct infringement of claim 1 is alleged in paragraphs 15 and 26. Paragraphs 16-22 and 27-33 then particularize the elements of the claims in Motorola's and Samsung's infringing products.

If this, in total, is not enough to put a defendant on notice that it directly and contributorily infringes, then no complaint can ever satisfy the pleading requirements of Rule 8.

II. LEGAL STANDARD

"Motions to dismiss test the sufficiency, not the merits, of the case." Groupon Inc. v. Mobgob LLC, 2011 U.S. Dist. LEXIS 56937, at *4 (N.D. Ill. May 25, 2011) (citing Gibson v. City of Chicago, 910 F.2d 1510, 1520 (7th Cir. 1990)). To survive a Rule 12(b)(6) motion to dismiss, a claim need only provide grounds for relief beyond "the speculative level." Groupon, 2011 U.S. Dist. LEXIS 56937, at *5 (citing Bell Atlantic v.

Twombly, 550 U.S. 544, 555 (2007)). “Specific facts are not necessary.” Id. at *5 (citing Erickson, 551 U.S. at 93). Importantly, in addressing a motion to dismiss, the court “treats well-pleaded allegations as true, and draws all reasonable inferences in the claimants’ favor.” Id.

There is no dispute that the standards enunciated in Bell Atlantic v. Twombly, 550 U.S. 544, 570 (2007) and Ashcroft v. Iqbal, 556 U.S. 662 (2009) apply to patent cases. But, there is also no dispute that this line of cases did not radically alter the pleading requirements of Federal Rule of Civil Procedure 8. As the Seventh Circuit held in Swanson v. Citibank, N.A.: “[c]ritically, in none of the three recent decisions [*Twombly* ... *Iqbal*] did the Court cast any doubt on the validity of Rule 8 of the Federal Rules of Civil Procedure. To the contrary: at all times it has said that it is interpreting Rule 8, not tossing it out the window.” 614 F.3d 400, 403 (7th Cir. 2010). Federal Rule of Civil Procedure 8(a)(2) requires only “a short and plain statement of the claim showing that the pleader is entitled to relief.” Specific facts are not necessary; rather, it is only necessary to “give the defendant fair notice of what the ... claim is and the grounds upon which it rests.” Erickson v. Pardus, 551 U.S. 89, 93-94 (2007) (citing Bell Atlantic Corp. v. Twombly, 550 U.S. 544, 555 (2007)). Here, Cascades has gone beyond mere notice -- it has shown when, where, why and how there is infringement.

III. THE THIRD AMENDED COMPLAINT SATISFIES THE PLEADING REQUIREMENTS OF THE FEDERAL RULES

Cascades’ 33 paragraph, 14-page Third Amended Complaint and attached 29 pages of claim charts clearly meets the Rule 8(a)(2) pleading standard. The suggested forms for filing complaints (Rule 84) “illustrate the simplicity and brevity that [the] rules contemplate.” Rule 84 was recently amended in 2007. Form 18 is an exemplary

Complaint for patent infringement that, according to the Rules, still satisfies the minimum notice requirement of the Federal Rules. It contains four paragraphs (not 33). It attaches no claim charts showing exactly why there is infringement. There is also no statement showing how each claim element is used by each accused product. The allegation of infringement says only this:

3. The defendant has infringed and is still infringing the Letters Patent by making, selling, and using electric motors that embody the patented invention, and the defendant will continue to do so unless enjoined by this Court.

(Form 18). By comparison, Cascades' Third Amended Complaint shows exactly why there is both direct and contributory infringement using illustrative claim charts, Exhibits A and B, as well as identifying Defendants' own officers and employees using accused infringing devices. Defendants' attempts to gloss over these demonstrations of use by their own personnel overlooks the fact that the Android operating system loaded onto each accused phone made, used and sold by Defendants operates inherently.

The Third Amended Complaint alleges more than making and selling with respect to method claim 15 -- it alleges that Motorola and Samsung have been **using** the claimed method with specifically identified products (D.I. 36, ¶¶ 12 and 23). Use of a method is an act of direct infringement. BMC Res., Inc., 498 F.3d at 1378. Separately, Motorola's and Samsung's manufacture and sale of their products is a direct infringement of claim 1 and that, too, is now alleged (D.I. 36, ¶¶ 15 and 26).

The Third Amended Complaint also alleges all the elements of a contributory infringement claim (D.I. 36, ¶¶ 13-14 and 24-25). There is notice of the patent; indeed, Judge Kendall's September 2011 Opinion in Trading Tech. Int'l, Inc. v. BCG Partners, Inc., 2011 U.S. Dist. LEXIS 99415, at *15-16 (N.D. Ill. Sept. 2, 2011), is on point with the facts of this case: (1) Motorola and Samsung received notice of its infringement through

the original complaint Cascades filed last July; (2) Motorola's and Samsung's infringement has continued despite such notice; and (3) Cascades' Third Amended Complaint alleges that Motorola and Samsung "knew and now knows" of their infringement. These indirect infringement allegations meet the level of pleading required by Trading Tech. Int'l.

Cascades also sufficiently pled that Motorola and Samsung intended that their customers use their products in an infringing way. The allegation that Motorola and Samsung sell the allegedly infringing products to their customers is enough to show specific intent. Id. at *16-17 ("[T]he Court may infer that by selling their infringing products the defendants intended their customers to use them and thus infringe [the plaintiff's] patents.").

Motorola and Samsung complain that Cascades' claim charts attached to its Amended Complaint (D.I. 23 at Exhs. A and B) showing their infringement rely on fragments of source code, that there is no allegation that the code is used, and that the mere existence of source code is not infringement. But, Cascades' December 15, 2011 Amended Complaint describes those illustrative products now known to infringe, the means by which infringement occurs, the conditions for infringement and the circumstances of infringement. (D.I. 23). This is now all explained in still more detail in paragraphs 14 and 25 of Cascades' Third Amended Complaint. (D.I. 36). Use of the method and, thus, direct infringement, occurs when a user activates the DROID³ or Nexus S phone which, in turn, activates the Dalvik Virtual Machine embedded in the phone. Id. The Dalvik Virtual Machine looks through the phone's applications, builds a tree of dependencies, which then optimizes the byte code for each application. Id.

Motorola and Samsung tout this as an advantage of their products that use the Android operating system.

As for Motorola/Samsung's contentions:

- claim charts rely on fragments of code: not so -- the Dalvik VM operation is explained in the complaint (see D.I. 36, ¶¶ 14 and 25);
- no allegations that Defendants or customers use code: not so -- see ¶¶ 13 and 24;
- simply making and selling a device is not direct infringement: apparatus claim 1 is directly infringed (see ¶¶ 15 and 26);
- no demonstration of how the accused products are used, by whom, and under what circumstances: not so (see ¶¶ 12, 14, 23 and 25).

Cascades' Third Amended Complaint more than adequately pleads infringement by Defendants.

IV. THE LAW REQUIRES FAR LESS DETAIL THAN CASCADES HAS PROVIDED

Nothing is "unspecified" in Cascades' Third Amended Complaint as Motorola and Samsung allege. Rather, specific acts of infringement are explained in detail. It is alleged that Motorola and Samsung perform each step of the Claim 15 method as demonstrated publicly by their own employees. Hence, the cases Motorola and Samsung cite on sale of equipment (see, e.g., Joy Techs) miss the point. Use is an act of direct infringement of a method claim. See BMC Res., Inc. v. Paymentech, L.P., 498 F.3d 1373, 1378 (Fed. Cir. 2007).

Further, this Court's decision in Avery Dennison actually supports Cascades. Here, there is not the mere possibility of misconduct. Specific use of each step of the method is alleged in detail both by Motorola and Samsung and their customers. The Defendants' fixation with cases dealing with making and selling (like Joy Techs) actually

supports Cascades as well, since the Third Amended Complaint not only alleges, but actually shows, how the claimed process is performed. The Local Patent Rules of this Court require a plaintiff to provide Initial Infringement Contentions with accompanying claim charts 28 days after a plaintiff's answer to a defendant's counterclaim has been filed. LPR 2.1 and LPR 2.2. Cascades provided a claim chart in its Amended Complaint (D.I. 23) months earlier than the Local Patent Rules require. And paragraphs 17, 19, 24 and 26 allege the elements of the asserted claims are present in Motorola's and Samsung's accused smartphones such as their DROID³ and Nexus S smartphones. (D.I. 36). All these allegations must be taken as true.

Defendants' reliance upon Federal Circuit authority concerning "the existence of source code alone" is markedly out of context. Both of these decisions arose from appeals in cases which were in completely different procedural stages than this case – after summary judgment (Ricoh) and after trial (Lucent Tech.), and so clearly after the initial pleading stage of a complaint and answer, and presumably after the completion of fact and expert discovery. In essence, Defendants are using these cases to require Cascades to **prove** infringement now, at the time it files its Complaint, and lays a completely different standard upon Cascades than is required by law. Even Defendants' own cases on pleading recognize that Cascades has met its burden with its Third Amended Complaint (and its earlier Complaint as well) – Adaptor cites the Federal Circuit's McZeal opinion, stating that for Cascades to sufficiently plead infringement by Defendants it can do so, as Cascades has already done, by "most likely **naming** the allegedly infringing product or method." Adaptor, Inc. v. Sealing Sys., 2010 WL 5463256, *2 (W.D. Wisc. Dec. 29, 2010). McZeal v. Sprint Nextel Corp., 501 F.3d 1354 (Fed. Cir. 2007) supports Cascades. As set forth by McZeal, "a patentee need only

plead facts sufficient to place the alleged infringer on notice as to what he must defend,” and is “not required to specifically include each element of the claims of the asserted patent.” McZeal, 501 F.3d at 1357. Specifically, Cascades’ pleading meets all of the requirements set forth by McZeal in its citation to Fed. R. Civ. P. Form 16 (now Form 18). As McZeal sufficiently pleaded a cause of action for infringement, so did Cascades’ Third Amended Complaint in this case: 1) assert plaintiff’s ownership of the patent-in-suit; 2) name Defendant Motorola and Defendant Samsung; 3) cite the patent-in-suit as allegedly infringed; 4) describe the means by which Defendants are allegedly infringing; 5) point to the specific patent law invoked; and 6) identify the allegedly infringing products manufactured and distributed by Defendants. See also, Mallinckrodt, Inc. v. E-Z-EM, Inc., 670 F. Supp. 2d 349, 353 (D. Del. 2009) (citing McZeal, 501 F.3d at 1356-57; Fed. R. Civ. P. Form 18 (2007)). In sharp contrast, Gharb was “completely void of a short and plain statement of a claim” and “makes no allegation that any product sold by or to any of the Defendants contains each limitation of the claims of the ‘654 patent.” Gharb v. Rockwell Automation, 2011 WL 5373989, *4-5 (N.D. Ill Nov. 4, 2011). As set forth above, Cascades’ Third Amended Complaint meets all of these requirements.

Aro, Global Tech and other cases cited by Defendants also miss the point: the Third Amended Complaint here pleads knowledge by Motorola and Samsung and direct infringement both by them and their customers.

V. CASCADES ADEQUATELY ALLEGED INFRINGEMENT OF CLAIM 1

Now Defendants’ argument essentially asserts that despite other defendants and their counsel having filed answers in response to Cascades’ allegation of direct infringement of the ‘750 patent, Cascades’ pleading is somehow inadequate with

respect to claim 1 as well. Yet, Cascades adequately pled direct infringement for system/apparatus claim 1 by Motorola's and Samsung's sale, offer for sale, importation or use of the Droid³ or Nexus S smartphones, among other Android-based products. (D.I. 36, ¶¶15 and 26). Cascades alleged that these accused devices contain each element of the claimed system. Contrary to Defendants' argument, infringement of a system or apparatus claim does not require any performance of steps or functions. See, e.g., Finjan, Inc. v. Secure Computing Corp., 626 F.3d 1197, 1204 (Fed. Cir. 2010); NTP, Inc. v. Research in Motion, Ltd., 418 F.3d 1282, 1318 (Fed. Cir. 2005) ("[T]he use of a process necessarily involves doing or performing each of the steps recited. This is unlike use of a system as a whole . . ."). Therefore, a user does not need to activate an accused product, nor does an error need to be encountered, in order to infringe claim 1. All that is required for infringement is that the structural elements or capabilities are present in the product.

In Finjan, the Federal Circuit held that an apparatus claim was infringed, even though the accused device needed to be programmed by the user for performing the claimed steps. Finjan, 626 F.3d at 1204. The patents at issue related to "practice scanning" technology for computer security. Id. at 1200. The accused products contained proactive scanning features that were locked when sold, requiring customers to purchase keys to unlock the proactive scanning modules. Nonetheless, the Federal Circuit found that the products directly infringed the apparatus claims. Specifically, the appeals court held:

The code for proactive scanning 'was already present' in Defendant's accused products when sold. There is no evidence that customers needed to modify the underlying code to unlock any software modules. The facts that users needed to 'activate the functions programmed' by

purchasing keys does not detract from or somehow nullify the existence of the claimed structure in the accused software.

Id. at 1205.

Similarly, in Fantasy Sports Props. v. Sportsline.com, the Federal Circuit provided that a sale of software would infringe on the asserted claims as long as it contained the means for performing the required functions, even if those functions required activation. Fantasy Sports, 287 F.3d 1108, 1118 (Fed. Cir. 2002) (“[A]n infringing software must include the ‘means for scoring . . . bonus points’ regardless whether that means is activated or utilized in any way.”). The appeals court held, “although a user must activate the functions programmed into a piece of software by selecting those options, the user is only activating means that are *already present in the underlying software*.” Id.

Defendants’ heavy reliance on Mikkelson Graphic Engineering Inc. v. Zund America, Inc., 2011 U.S. Dist. 141548 (E.D. Wis. Dec. 8, 2011), is inapplicable to this case. In Mikkelson, the defendant sold systems with software that infringed apparatus claims of the plaintiff’s patent only when a search function was enabled. The district court considered whether software sold by the defendant with that search functionality disabled still infringed the claims. The Mikkelson defendant disabled the search function by modifying the configuration files so that a computer running the software would ignore the part of the source code instructing it to perform a search operation. Id. at *9. Users were required to reconfigure the software in order for it to operate in an infringing manner. Id. at *15. The district court found that the system did not contain the infringing elements because the configuration files were part of the structure of the accused device, and thus the claimed structure was not present in the device at the

time of sale. Id. (“[T]he configuration files are part of the accused device, and so if the user modifies the configuration files, the user is modifying the structure of the device.”)

Here, Cascades has adequately pled that Defendants’ cell phones and tablets contain the infringing elements as those products are manufactured, sold, offered for sale, and imported; Defendants do not deny this. Unlike the Defendants’ offered authority Mikkelson, the user of Defendants’ cell phones or tablets is not required to make any modifications to the structure of the products in order for them to infringe claim 1. As provided in Cascades’ Third Amended Complaint, the Defendants’ accused products already possess each element or capability listed in the claim. (D.I. 36, ¶¶15 and 26).

VI. CONCLUSION

In sum, there is no basis to dismiss Cascades’ Third Amended Complaint and Cascades respectfully requests that the Court deny Motorola’s and Samsung’s Motion to Dismiss. In Case No. 11-cv-7264 before Judge Kennelly, defendant Dell, also represented by the same counsel as Motorola in this case, withdrew its pending dismiss motion in the face of a similarly detailed amended complaint filed by Cascades, and HTC made the same decision in Case No. 11-cv-6235 before Judge Guzman. Accused infringers LG, Pantech, Acer, and Sharp have each also found Cascades’ infringement pleadings sufficient to file an Answer. Motorola and Samsung should be instructed to do the same here.

Respectfully submitted,

/s/ Paul C. Gibbons

Raymond P. Niro (miro@nshn.com)

Arthur A. Gasey (gasey@nshn.com)

Paul C. Gibbons (gibbons@nshn.com)

Anna B. Folgers (afolgers@nshn.com)

NIRO, HALLER & NIRO

181 W. Madison, Suite 4600
Chicago, IL 60602
Attorneys for
CASCADES COMPUTER
INNOVATION, LLC

CERTIFICATE OF SERVICE

The undersigned hereby certifies that on March 15, 2012 the foregoing **CASCADES' OPPOSITION TO MOTOROLA'S AND SAMSUNG'S MOTION TO DISMISS** was filed electronically with the Clerk of the Court for the Northern District of Illinois using the Court's Electronic Case Filing System, which will send notification to the registered participants of the ECF System as listed:

Motorola Mobility:

Jonathan E. Retsky
Imron T. Aly
Kathleen B. Barry
WINSTON & STRAWN LLP
35 West Wacker Drive
Chicago, IL 60601
Tel: (312) 558-5600
Fax: (312) 558-5700
jretskey@winston.com
ialy@winston.com
kbarry@winston.com

Samsung Electronics:

Christian Chadd Taylor
KIRKLAND & ELLIS
555 California Street
27th Floor
San Francisco, CA 94104
415-439-1823

I certify that all parties in this case are represented by counsel who are CM/ECF participants.

/s/ Paul C. Gibbons
Attorney for Cascades Computer Innovation,
LLC

Exhibit F

IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF ILLINOIS
EASTERN DIVISION

CASCADES COMPUTER INNOVATION, LLC,)	
)	
Plaintiff,)	
)	
vs.)	No. 11 C 7264
)	
DELL INC.,)	
)	
Defendant.)	
<hr/>)	
CASCADES COMPUTER INNOVATION, LLC,)	
)	
Plaintiff,)	
)	
vs.)	No. 11 C 4574
)	
MOTOROLA MOBILITY HOLDINGS, INC., and SAMSUNG ELECTRONICS CO., LTD.)	
)	
Defendants.)	
<hr/>)	
CASCADES COMPUTER INNOVATION, LLC,)	
)	
Plaintiff,)	
)	
vs.)	No. 11 C 6235
)	
HTC CORPORATION and LG ELECTRONICS, INC.,)	
)	
Defendants.)	
<hr/>)	

CASCADES COMPUTER INNOVATION, LLC,)	
)	
Plaintiff,)	
)	
vs.)	No. 11 C 7223
)	
SONY ERICSSON MOBILE COMMUNICATIONS (USA) INC.,)	
)	
Defendant.)	
<hr/>		
CASCADES COMPUTER INNOVATION, LLC,)	
)	
Plaintiff,)	
)	
vs.)	No. 11 C 7249
)	
PANTECH WIRELESS, INC.,)	
)	
Defendant.)	

MEMORANDUM OPINION AND ORDER

MATTHEW F. KENNELLY, District Judge:

In separate lawsuits, Cascades Computer Innovation, LLC, has sued several defendants for patent infringement, contending that the defendants manufacture products that infringe U.S. patent number 7,065,750 (the '750 patent). Each of these claims was consolidated before the undersigned judge pursuant to Northern District of Illinois Internal Operating Procedure 13 for purposes of claim construction and, via a subsequent order, for consideration of parallel motions to dismiss filed in each of the cases. The Court previously denied the motions to dismiss, *see Cascades Computer Innovation, Inc. v. Motorola Mobility Holdings, Inc.*, Nos. 11 C 4574, 11 C 7223, 11 C

7249, 11 C 7252 & 11 C 7264, 2013 WL 3366276 (N.D. Ill. July 5, 2013), and now rules on the construction of the disputed terms in the '750 patent.

Background

The '750 patent is entitled "Method and Apparatus for Preserving Precise Exceptions in Binary Translated Code." See Defs.' Ex. 1 (Patent). The issue date of the patent was June 20, 2006, and the inventors listed are four individuals from Russia, who first filed the application for the patent in 1999. Cascades, the plaintiff in this case, is the exclusive licensee under the patent.

In general terms, the patent describes a method for efficiently executing on one system architecture computer programming code that is intended for a different architecture. The introductory paragraph of the "Background of the Invention" section of the '750 patent's specification describes the invention this way: "The present invention relates to a computer system executing foreign code and more particularly to a computer system and method for efficient handling of exceptions that arise when executing binary translated code." Patent at 1:36–39. The patent contains eighteen claims, two of which are at issue. Claim 1 describes the invention's binary translation system, listing six elements; Claim 15 describes a recomputing method with three elements. The parties dispute the definitions of eight terms within these two claims. For seven of the eight terms, Cascades argues that the Court need not make any construction, because the plain and ordinary meaning would be apparent to "one of ordinary skill" in the software programming field. The defendants, on the other hand, offer language from the specification for most of their proposed definitions, often arguing that the patentees sought to expressly define these terms.

Discussion

1. "Foreign code"

The term "foreign code" appears in both Claims 1 and 15 of the '750 patent. Claim 1 refers to "a non-optimizing foreign code execution module configured to maintain dedicated foreign state for each foreign binary operation executed allowing for the exceptions arisen to be handled precisely." Patent at 16:6–9. The term appears again near the end of Claim 1, where the patent states that the invention's recovery mechanism is configured "to continue foreign codes [sic] execution in case of the exception arisen during the execution of the corresponding optimized host codes." *Id.* at 16:32–35. The term also appears three times in Claim 15, each dealing with "translation of the foreign code." *Id.* at 17:23–24, 26–27, 29–30. The specification provides an express definition for the term "foreign binary code," when it states that, "[a]s used herein, foreign binary code means computer instructions written for execution on a foreign processing system but ported to the host computer system 100." *Id.* at 6:49–52.

The defendants' proposed construction of the term foreign code is: "computer instructions written for execution on a foreign processing system but ported to the host computer system." See Joint Notice of Modified Proposed Claim Constructions at 3 [docket no. 123]. They argue that the specification "expressly defined" the term "foreign code" to include this entire phrase. Hrng. Trans. at 39–40. Cascades argues in favor of the "plain and ordinary meaning" of this term as "understandable by one of ordinary skill in the art." See Joint Notice of Modified Proposed Claim Constructions at 3 [docket no. 123]. If, however, the Court determines to construe the term, Cascades proposes the

following definition: "computer instructions written for execution on a foreign processing system." *Id.* This is the same as the defendants' construction minus the last seven words, which Cascades argues are "just not necessary." Hrng. Trans. at 41. Plaintiff contends that it is obvious the patent requires porting between systems and that "porting is not part of the code," and also that the phrase at issue "is a pretty common term" among computer programmers. *Id.* at 41–43. The Court notes that at the claim construction hearing, attorneys for both sides also agreed that the Court's choice of one of these interpretations over the other will make no difference in the ultimate resolution of this dispute.¹ See *id.* at 42–43.

The Court agrees with Cascades that construction of this term is not necessary and that its meaning would be readily apparent to a computer programmer of ordinary skill. It does not take the expertise of a software engineer to determine that "code" is a bedrock concept in the field of computer programming. As the Federal Circuit has repeatedly stressed, "[c]laim terms generally are construed in accordance with the ordinary and customary meaning they would have to one of ordinary skill in the art in light of the specification and the prosecution history." *SanDisk Corp. v. Kingston Tech. Co.*, 695 F.3d 1348, 1353 (Fed. Cir. 2012). The Court does not have trouble concluding that a typical computer programmer reading the specification would understand what the claims mean by "code," and that the references to "foreign code" clearly denote code written for a system other than the host system. The claims discuss "foreign code"

¹ The parties had disagreed in their briefs over whether defendants draw their definition from the specification's definition of the correct term; Cascades contended that "foreign binary code" is not the same thing as "foreign code." But at the hearing, Cascades adopted an interpretation that uses much the same language as the defendant's construction.

and "foreign operations" in conjunction with "host operations" and "corresponding optimized host codes." See, e.g., Patent at 16:10–14; 16:32–34. These terms provide the context, if it were not already apparent to a programmer, that the "foreign code" discussed in the claims is code meant for another system; further, as Cascades pointed out at the claim construction hearing, the patent's whole reason for being is to "port[] [code] to the host computer system." See *id.* at 6:49–52.

It is true that the patent says "foreign binary code *means* computer instructions written for execution on a foreign processing system but ported to the host computer system 100." *Id.* It is also true that the Federal Circuit has stated that "the inventor's lexicography governs" in cases where "the specification . . . reveal[s] a special definition given to a claim term by the patentee." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1316 (Fed. Cir. 2005) (en banc). Such a definition, however, governs only when the patentee's definition for the term "differs from the meaning it would otherwise possess." *Id.* Neither party has argued that the long definition given to "foreign binary code" in the patent is different from what an average software programmer would have understood. As the Federal Circuit stated in *Phillips*, "[t]he inquiry into how a person of ordinary skill in the art understands a claim term provides an objective baseline from which to begin claim interpretation." *Id.* at 1313. In the case of the term "foreign code" in the context of software engineering, the Court determines there is no need to go beyond that baseline and therefore declines to construe this term.

2. "Binary translation/translator"

Claim 1 of the patent describes a "binary translation system" including "an optimizing binary translator configured to translate foreign binary operations into

optimized sequences of host operations." Patent at 16:5, 16:10–12. The term "binary translation" appears again in Claim 15, which describes a "method of recomputing a dedicated foreign state in a binary translation system from documentation generated by an optimizing translator in a case of an exception arising during execution of optimized binary translated code translated from a foreign code." *Id.* at 17:20–24. Claim 15 makes two references to "optimized binary translated code" in which the invention renders recovery points. *Id.* at 17:25–28, 31–35.

This is the sole disputed term for which Cascades does not initially advocate for the Court's adoption of its purported plain and ordinary meaning. Instead, Cascades contends that the correct definition of "binary translation" is "foreign code is processed by host software to produce new host code corresponding to the foreign code." Pl.'s Resp. at 7. Cascades argues this is "an explicit definition" from the patent itself. *Id.* It points to a passage from the specification that says "[b]inary translation means that a foreign code is processed by host software to produce new host code corresponding to the foreign code." Patent at 4:16–18. At the claim construction hearing, the defendants agreed that the Cascades construction "does come right out of the specification," but they argued that the phrase "corresponding to" in that definition adds unhelpful ambiguity to the term and "reads out" the translation element of the term. Hrng. Trans. at 44–45. Therefore, the defendants contend, their proposed definition goes further in that it helps explain what "corresponding to" means in the context of "binary translation." They propose this interpretation: "generation/generator of a new sequence of host code that performs the same functions and achieves the same behavior as on the foreign platform." Defs.' Br. at 6–7. This language, too, is drawn from the specification, in a

section that defendants say serves to "refine[]" the "general notion of 'binary translation' by expressly defining the term." *Id.* at 7. Cascades countered at the hearing that the defendants' interpretation does not actually define the term and that a typical software engineer would know what "corresponding to the foreign code" means.

In construing the terms of a claim, courts must "first look to, and primarily rely on, the intrinsic evidence, including the claims themselves, the specification, and the prosecution history of the patent, which is usually dispositive." *Sunovion Pharms., Inc. v. Teva Pharms. USA, Inc.*, 731 F.3d 1271, 1276 (Fed. Cir. 2013). Courts are "generally" to give the words of a claim "their ordinary and customary meaning." *Phillips*, 415 F.3d at 1312–13. However, an "exception[]" to this rule applies "when a patentee sets out a definition and acts as his own lexicographer." *Thorner v. Sony Computer Ent. Am. LLC*, 669 F.3d 1362, 1365 (Fed. Cir. 2012) (citing *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1580 (Fed. Cir. 1996)); *see also Vitronics*, 90 F.3d at 1582 ("The specification acts as a dictionary when it expressly defines terms used in the claims or when it defines terms by implication.").

The defendants face two difficulties in advocating for their proposed definition of this term. First, the patent directly provides a meaning for "binary translation" in one of the few explicit definitions in the entire document. The patent specifically states that "[b]inary translation *means* that a foreign code is processed by host software to produce new host code corresponding to the foreign code." Patent at 4:16–18 (emphasis added). The Court has little trouble concluding that this is precisely what *Vitronics* is referring to when it says that "the specification acts as a dictionary when it expressly defines terms used in the claims." *Vitronics*, 90 F.3d at 1582. The defendants are

attempting to add to this patent-provided definition with their construction of the term, but *Phillips*, *Vitronics*, and similar cases do not suggest that a party should be permitted to add to an express definition provided in the patent. The defendants do not cite cases indicating that this is appropriate. Second, the defendants' definition comes from a passage of the specification stating what binary translation *may achieve*, not what it *is*. Here is that passage in full: "Yet another technique is to use binary translation to generate a sequence of instructions that perform the same functions and achieve the same behavior as on the foreign platform." Patent at 2:1–4. Describing a set of results from binary translation is different from stating what the term "means."

Given the direct definition of "binary translation" found in the specification of the '750 patent, the Court construes the disputed claim language as advocated by Cascades: "foreign code is processed by host software to produce new host code corresponding to the foreign code." Pl.'s Resp. at 7.

3. "Optimized/optimizing"

The term "optimize" or "optimizing" appears, in one form or another, a dozen times in the two claims at issue. In Claim 1, the described binary translation system includes both "a non-optimizing foreign code execution module" and "an optimizing binary translator configured to translate foreign binary operations into optimized sequences of host operations." Patent at 16:6, 10–12. The same claim describes a "documentation generator configured to generate a set of documentations for optimized sequences of host operations," and it also refers to a recovery mechanism that is supposed to "continue foreign codes execution in case of the exception arisen during the execution of the corresponding optimized host codes." *Id.* at 16:15–17, 26–34. The

references to the term in Claim 15 are similar, in describing an "optimizing translator" or "optimized translation," as well as "optimized binary translated code." *Id.* at 17:22–24.

The defendants propose that to "optimize" means to "extract the inherent parallelism of the foreign code." Defs.' Br. at 8. They draw this definition from the specification. See, e.g., Patent at 4:24–25 ("Optimization requires extracting the inherent parallelism of the foreign code."); *id.* at 10:61–64 ("The optimizing process extracts the parallelism inherent to foreign code . . ."). Also, the reference to two types of translators in the patent supports this construction, the defendants say. Though the optimizing translator has a parallel approach, they argue, the non-optimizing translator "eliminates the optimization by re-executing the code in a *sequential* manner" (as opposed to parallel). Defs.' Br. at 10 (citing Patent at 15:55–57). At bottom, the defendants contend, "optimization" in the '750 patent means "that the code be made to run in parallel." Defs.' Repl. at 4. At the claim construction hearing, the defendants argued that "the only optimization in this patent is [this] parallelism." Hrng. Trans. at 57. "[W]henever [the patent] refers to optimization," the defendants argued, "what it's really talking about is when it's doing that binary translation, it's putting that code . . . into parallel." *Id.* at 59. They contended that although the patent does not expressly define "optimize" in the way they propose, it does so by implication, citing *Honeywell Int'l, Inc. v. ITT Indus., Inc.*, 452 F.3d 1312, 1318 (Fed. Cir. 2006), and *C.R. Bard, Inc. v. United States Surgical Corp.*, 388 F.3d 858, 864 (Fed. Cir. 2004).

Cascades argues that the term "optimize" need not be defined for purposes of claim construction. It argued at the hearing that the word is "a well-understood term to a lay jury that is also consistent with what one of ordinary skill in the art would also

understand it to mean." Hrng. Trans. at 67–68. Cascades argues for the ordinary meaning of the term, or a generalized definition, because the word "has no particular or specialized meaning within the relevant art and is well within the understanding of a normal jury." Pl.'s Resp. at 9. To optimize, Cascades argues, is to "improve performance to run efficiently." *Id.* It cites two definitions of optimization from the American Heritage dictionary: "[t]o make as perfect or effective as possible," and "[t]o increase the computing speed and efficiency of (a program), as by rewriting instructions." Pl.'s Ex. A. It also cites a definition of the term "optimizing compiler" from Webster's 1997 *Dictionary of Computer Terms*. See Pl.'s Ex. B (a machine "that translates source code into machine language optimized to run as efficiently as possible on a particular microprocessor" (alterations omitted)). This, Cascades says, indicates that the definition of optimize is the same among laypeople and those of ordinary skill in computer science. Cascades proceeds to point to several passages in the patent specification where optimization is mentioned in conjunction with improving performance, and it points out that the specification uses "optimizing" to describe other processes as well. See, e.g., Patent at 3:59–62 ("The present invention includes a means for optimizing the execution of binary translated code by reordering of the execution order of pending operations including memory access operations."). Cascades adds that defendants' proposed construction—"extract the inherent parallelism of the foreign code"—is a phrase that could confuse a jury.

As the Federal Circuit has observed, "[i]n some cases, the ordinary meaning of claim language as understood by a person of skill in the art may be readily apparent even to lay judges, and claim construction in such cases involves little more than the

application of the widely accepted meaning of commonly understood words." *Phillips*, 415 F.3d at 1314. Such terms "are generally given their ordinary and customary meaning" according to "a person of ordinary skill in the art in question at the time of the invention." *Id.* at 1312–13. However, "[i]diosyncratic language, highly technical terms, or terms coined by the inventor are best understood by reference to the specification." *3M Innovative Props. Co. v. Tredegar Corp.*, 725 F.3d 1315, 1321 (Fed. Cir. 2013). In addition, a patentee need not expressly define a term in order for a definition of that term to be drawn from the patent. Instead, a definition "may be inferred from clear limiting descriptions of the invention in the specification or prosecution history." *Aventis Pharma S.A. v. Hospira, Inc.*, 675 F.3d 1324, 1330 (Fed. Cir. 2012). Yet courts must take care not to infer that a patentee intended to define a term merely because only a single example or embodiment of the term is described in the patent itself. The Federal Circuit "has expressly rejected the contention that if a patent describes only a single embodiment, the claims of the patent must be construed as being limited to that embodiment." *Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 906 (Fed. Cir. 2004) (collecting cases). The Federal Circuit recently had the occasion to again address how courts are to navigate this area of claim construction:

It is . . . not enough that the only embodiments, or all of the embodiments, contain a particular limitation. We do not read limitations from the specification into claims; we do not redefine words. Only the patentee can do that. To constitute disclaimer, there must be a clear and unmistakable disclaimer.

Thorner, 669 F.3d at 1366–67.

The cases the defendants cite stand for the proposition that a definition can, in certain circumstances, be taken from a specification even if it does not expressly define

the particular term. In *Honeywell*, the Federal Circuit held that limitation of the term "fuel injection system component" to mean a fuel filter (and nothing else) was appropriate. *Honeywell*, 452 F.3d at 1318. The court concluded that the concept of a fuel filter "was not merely discussed as a preferred embodiment" in the specification because "the written description refer[red] to the fuel filter as 'this invention' or 'the present invention'" at least four times. *Id.* (citing such language as, "This invention relates to a fuel filter for use in the fuel line that delivers fuel to a motor vehicle engine."). This contextual evidence helped convince the court that the fuel filter was the only embodiment of the claim term in the invention. Only one other fuel component was mentioned in the description (a fuel line), but that component "was not required by the patentee to be made of an electrically conductive polymer material, as the claims require." *Id.* Given these facts, it was proper "to take the patentee at his word and the word was that the invention is a fuel filter." *Id.*

In *C.R. Bard*, the term at issue referred to a medical device that included a plug. *C.R. Bard*, 388 F.3d at 860. The "Summary of the Invention" section of the patent specification stated that the invention was "an implantable prosthesis," then went on to say that "[t]he implant includes a pleated surface." *Id.* Similarly, the patent's abstract referred to "[a]n implantable prosthesis including a conical mesh plug having a pleated surface." *Id.* at 860–61. The court concluded that although the patent did not expressly define the plug, these statements "unequivocally define[d] the claimed plug as having pleats" in two places where it "describe[d] in general terms what it deem[ed] to be the invention." *Id.* at 864. The Federal Circuit found the placement of these statements helpful in determining their significance. Although placement of a description in that

section of a patent is not "determinative," it "can signal the likelihood that the statement will support a limiting definition of a claim term." *Id.* However, the court cautioned that the weight given to such language "must, of course, be determined on a case-by-case basis." *Id.* In *C.R. Bard*, there were "[s]tatements of general applicability" that "clearly define[d]" the plug as having pleats, which was the end of the matter. *Id.* at 866.

In the present case, the difficulty with the defendants' interpretation is that they draw it from passages of the patent describing the end result of optimization, and not what optimization, as a process, actually involves. There is no direct description of the process of optimization, as there was for the clear definition of the plug in *C.R. Bard* or the concrete characterization in *Honeywell*. For example, the first passage the defendants label as definitional in their brief ("[o]ptimization requires extracting the inherent parallelism of the foreign code," Patent at 4:24–25) does not actually say what optimization is or what it does; rather, the passage merely says what optimization requires. A statement of what a process requires is not the same as defining that process; for example, doing homework might require putting pen to paper, but identifying that function does not provide a definition of what homework is. The same is true of the other passages the defendants cite, e.g., "[t]o exploit the explicit parallelism of this architecture . . . it is necessary to optimize the binary translated code in a manner that maintains precise exceptions." *Id.* at 8:61–64. And indeed, one passage of the specification makes clear that optimization and extracting parallelism are two different things, as one can produce the other: "[T]o exploit parallelism of the host processor architecture in binary translated code, the host code must be optimized." *Id.* at 4:22–24.

It therefore seems fairly clear that the patentees did not seek to expressly define the term "optimize" and that the passages defendants cite cannot be used for that purpose. Though the term is mentioned multiple times in conjunction with extracting or exploiting the parallelism of a particular system architecture, it does not follow that such extraction or exploitation is the only means of optimizing, nor that the patentee intended it to be. The defendants may be right that running translated code in parallel is the *sine qua non* of this particular patent, see Defs.' Repl. at 5, but that does not mean that "optimize" has the specific definition they wish to assign it. As Cascades points out, "optimize" is not always mentioned in direct conjunction with exploiting parallelism, but rather with improvement of performance. In one of the embodiments, the "optimizing binary translation process" is described as one intended "to improve performance"; no mention is made of code running in sequence or parallel. Patent at 7:12–15; see *also id.* at 7:56–59. Another passage states that *performance* can be optimized, which further argues against adopting the defendants' specific definition: "to optimize performance, load operations can be moved ahead of store operations." *Id.* at 7:22–23.

The Court concludes that the defendants have not offered sufficiently compelling evidence that the patentees established a "lexicography" encompassing this term that differs from its plain meaning as understood in the computer programming field. The Court concludes that the term "optimizing" in the '750 patent, though used frequently in association with extracting parallelism, is not defined by that function, nor is it "[i]diosyncratic," "highly technical," or "coined by the inventor," *3M Innovative Props.*, 725 F.3d at 1321. Rather, the term requires no construction beyond its ordinary and customary meaning to a person in the art at the filing date of the patent, see *Phillips*,

415 F.3d at 1312–13—that is, to make the most effective use of, or to make most efficient.

4. "Documentation"

Apart from the disputed phrase "documentation generator configured to generate," discussed below, the term "documentation" (or a variant) appears fifteen times in the claims at issue. It is used as both a singular and plural noun, sometimes in the same sentence. For example, Claim 15 describes a recomputing method that "generat[es] a set of documentations during the optimized translation of the foreign code, wherein each documentation in the set of documentations corresponds to a recovery point in the optimized binary translated code" Patent at 17:29–32. Claim 1 refers to "documentation" as an indefinite noun: "for each host operation address, operations required to calculate a corresponding foreign state for the host operation address are added to documentation." *Id.* at 16:22–25. Claim 15 also describes part of the invention as a "method of recomputing a dedicated foreign state in a binary translation system from documentation generated by an optimizing translator." *Id.* at 17:20–22. "Documentation" also serves as a modifier; in Claim 1, there is reference to "a documentation tracker configured to record host operation addresses at appointed points of the host operation sequences being executed," *id.* at 16:20–22, a function that differentiates the tracker from the "documentation generator."

The specification states that "[e]very recovery point is described by a documentation set . . . that contains information where all foreign registers are located in the host registers in the optimized binary translated code." Patent at 9:35–41. The defendants draw from this passage nearly verbatim for their proposed definition of the

term documentation: "data describing where all foreign registers are located in the host registers in the optimized binary translated code." Joint Notice of Modified Proposed Claim Constructions at 2 (docket no. 123). They cite similar language in Claim 1, see Patent at 16:15–19, and Claim 15, see *id.* at 17:30–34. At the claim construction hearing, the defendants argued that the definition of "documentation" cannot be "generic[]," because a generic definition of the term "may not be enough information within the context of the patent to recreate that foreign state which is essential to move on in the program." Hrng. Trans. at 71. The defendants contended that it is essential for the definition to reflect that a single documentation contains information about *all* registers related to a single recovery point. Furthermore, the defendants argued the patent actually imposes two requirements on a documentation: that it include "a list of registers with values in them," but also "operations to calculate the foreign state using those values." *Id.* at 81.

At the claim construction hearing, Cascades contended that this term requires no construction. That is because "the claim itself tells you what documentation is," Cascades argued, pointing to passages from both Claims 1 and 15. *Id.* at 77. The passage Cascades cites from Claim 1 states that "each documentation describes operations required to calculate a corresponding foreign state for an appointed point." Patent at 16:17–19. The cited passage from Claim 15 says that "each documentation . . . corresponds to a recovery point in the optimized binary translated code and describes operations required to calculate a corresponding foreign state for the recovery point." *Id.* at 17:30–34. Cascades argues that defendants' proposal is actually a definition for a documentation *set*, not a *documentation*. It is incorrect,

Cascades argues, to define the term as data describing the location of *all* foreign registers, because each documentation "only requires calculating a foreign state for a given, 'appointed point' (in the case of claim 1)." Pl.'s Resp. at 12. (Cascades did not, however, explain how a "documentation set" is different from a "documentation.")

Defendants reply that Cascades conflates the plural "set of documentations" with the singular "documentation set" or "set of documentation." "When discussing a plurality of documentations, the '750 patent consistently calls that plurality a 'set of documentations.'" Defs.' Repl. at 8. Elsewhere, the patent uses both "documentation" and "documentation set" "to describe a single documentation associated with a single Recovery Point that describes the location of the foreign registers associated with that Recovery Point." *Id.* at 7. Defendants also argue that the plain and ordinary meaning of "documentation" is inappropriate for purposes of constructing the claims at issue, because the patent "describes a very specific and proprietary 'documentation.'" *Id.* at 9.

Neither party's proposal appears to capture the meaning of the term, which as described above is used in different ways in the patent. Defendants seem to want to narrow what would appear to be a general term to a particular contextual usage. Though the defendants draw from the specification for their definition, there are other uses of the term in the patent that are not as particularized as the passage the defendants use. But Cascades' approach, which is to say that no construction is needed, is not particularly helpful given the varying usages of the term. Cascades argued at the claim construction hearing that no interpretation is needed because the claims themselves define "documentation." As indicated, Cascades cites a passage from Claim 1 stating that "each documentation describes operations required to

calculate a corresponding foreign state for an appointed point," Patent at 16:17–19, and a passage from Claim 15 stating that "each documentation . . . corresponds to a recovery point in the optimized binary translated code and describes operations required to calculate a corresponding foreign state for the recovery point." *Id.* at 17:30–34. But neither of these passages defines documentation; rather, they describe what certain documentation does in particular situations.

The Court has returned to the patent for guidance on what the inventors meant by the term and how a person of ordinary skill in the field would have understood it. The specification contains two examples of "documentation," which help illustrate what it is. The specification states in columns 13 and 14:

There are three Recovery Points in the code. The first one describes register contents before starting execution. The second point (second wide instruction) is described by the following documentation:

Foreign register	Host register
EDX	R1
ECX	R2
ESI	(not changed)
FLAGS	R3

The second point is described by this documentation because registers R1, R2 and R3 have been released by the host optimizing scheduler and then reused in further calculations. After finishing execution of the code the documentation will have the following contents:

Foreign register	Host register
EDX	R1
ECX	R2
ESI	R4
FLAGS	R5

Patent at 13:50-67 & 14:1-8.

As shown by these illustrative examples, the term "documentation" has a far more generalized meaning than the very specific one that defendants seek to ascribe to it. The Court rules that documentation, as used in the present context, simply means written text containing information that describes a software operation. Further particularization of the type, nature, or contents of any given documentation is provided by the claim language or from—for example, in the fourth element of claim 1, the documentation is said to "describe[] operations required to calculate a corresponding foreign state for an appointed point." Patent at 16:17-19. It would be both unnecessary and inappropriate to load this sort of particularization into the definition of the term "documentation" itself.

5. "Documentation generator configured to generate"

The term "documentation generator configured to generate" appears in Claim 1 of the patent, where it is listed among the elements included in the patent's binary translation system. The claim says that the system comprises "a documentation generator configured to generate a set of documentations for optimized sequences of host operations, wherein each documentation describes operations required to calculate a corresponding foreign state for an appointed point." Patent at 16:15–19.

For this term, neither party offers a specific definition. The defendants argue that "documentation generator configured to generate" is a "means-plus-function term" within the meaning of 35 U.S.C. § 112(f). They contend that because the term itself evokes no specific structure, and because structure for the term cannot be found in the specification, the term is indefinite, and thus the claim containing it (Claim 1) is invalid. In their opening brief, the defendants implied that the patent should have included some

sort of algorithm for the documentation generator, and they expanded on this argument at the hearing, arguing that algorithms should receive "some degree of heightened scrutiny." Hrng. Trans. at 87. To support this notion, they cited *Aristocrat Techs. Australia Pty Ltd. v. Int'l Game Tech.*, 521 F.3d 1328, 1334 (Fed. Cir. 2008), a case where the claim's description of the structure for a computer-implemented function went "no farther than saying that the claimed functions are performed by a general purpose computer" and was thus invalid. There, the Federal Circuit noted that plaintiffs need not "produce a listing of source code or a highly detailed description of the algorithm to be used to achieve the claimed functions in order to satisfy 35 U.S.C. § [112(f)]." *Id.* at 1338. But they must "at least disclose the algorithm that transforms the general purpose microprocessor to a special purpose computer programmed to perform the disclosed algorithm." *Id.* (internal quotation marks omitted). At the hearing, the defendants also cited *Ex Parte Rodriguez*, No. 2008-000693, slip op. at 20–27 (B.P.A.I. Oct. 1, 2009), *available at* <http://www.uspto.gov/ip/boards/bpai/decisions/prec/fd080693.pdf>, a case from the Board of Patent Appeals and Interferences, in which claims containing the term "system configuration generator" were found invalid because the term was indefinite.

Cascades says this term should be given its plain and ordinary meaning as understandable by one of ordinary skill in this field. Cascades also argues that the burden to establish indefiniteness "by clear and convincing evidence" belongs to defendants, "including [when arguing] a lack of alleged structure." Pl.'s Resp. at 16 (citing *Budde v. Harley Davidson, Inc.*, 250 F.3d 1369, 1376 (Fed. Cir. 2001)). Because "[d]efendants have not provided any evidence" about the understanding of a "person of

skill" with regard to a "documentation generator," Cascades says, they have not met their burden to prove that the term is indefinite. *Id.* At the claim construction hearing, Cascades argued that the Court therefore does not have to consider whether the patent provides an algorithm for the "documentation generator" term in accordance with the *Aristocrat* case. Cascades also argues that the specification sufficiently describes structure for a documentation generator, an argument that the Court will address in detail below.

Under 35 U.S.C. § 112(f), formerly known as section 112, paragraph 6, "[a]n element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof." The statute then states that "such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof." *Id.* The Federal Circuit has stated that the application of section 112(f) is a "straightforward" question. *Inventio AG v. ThyssenKrupp Elevator Am. Corp.*, 649 F.3d 1350, 1356 (Fed. Cir. 2011). If a claim term contains the word "means," a rebuttable presumption that section 112(f) governs construction of the term is triggered; if the word "means" is absent, a presumption is triggered that section 112(f) does not govern. *Id.* This latter presumption is "a strong one that is not readily overcome." *Lighting World, Inc. v. Birchwood Lighting, Inc.*, 382 F.3d 1354, 1358 (Fed. Cir. 2004). The Federal Circuit recently put this presumption in strong terms: "When the claim drafter has not signaled his intent to invoke § [112(f)] by using the term 'means,' we are unwilling to apply that provision without a showing that the limitation essentially is

devoid of anything that can be construed as structure." *Flo Healthcare Solutions, LLC v. Kappos*, 697 F.3d 1367, 1374 (Fed. Cir. 2012).

In evaluating whether this presumption should stand, "it is sufficient if the claim term is used in common parlance or by persons of skill in the pertinent art to designate structure, even if the term covers a broad class of structures and even if the term identifies the structures by their function." *Lighting World*, 382 F.3d at 1359–60.

"Ultimately, whether claim language invokes § [112(f)] depends on how those skilled in the art would understand the structural significance of that claim language, assessed against the presumptions that flow from a drafter's choice to employ or not employ the term 'means.'" *Inventio*, 649 F.3d at 1360. Courts may "review[] the intrinsic record, as well as extrinsic evidence such as technical dictionaries, to determine if the challenger successfully rebutted the presumption that a claim that lacks the term 'means' is not subject to § [112(f)]." *Id.* at 1357. In terms of the burden a challenging party must meet to prove indefiniteness in the context of a means-plus-structure dispute, "[t]he party alleging that the specification fails to disclose sufficient corresponding structure must make that showing by clear and convincing evidence." *TecSec, Inc. v. IBM Corp.*, 731 F.3d 1336, 1349 (Fed. Cir. 2013).

To sum up, the law presumes that the term "documentation generator" is not subject to section 112(f) because the term does not use the word "means." To overcome the presumption, defendants' burden is to provide evidence that a typical software programmer would not understand that the term documentation generator connotes structure. The Court acknowledges that the term "documentation generator" is not directly defined in the patent, nor is it particularly revealing to the uninitiated. But

the defendants nonetheless have not produced persuasive evidence, as is their burden, that a typical software engineer would not understand what structure lies behind a documentation generator as included in the '750 patent. This is so in light of Cascades' contention that there is some structure for the generator in the patent and its allusions to knowledge in the relevant field.

In this case, the term appears only once in the entire patent, in one of the disputed claims. At no time does the specification otherwise reference a "documentation generator," although there are, as discussed above, many references to "documentation." In light of the absence of a definition for the term in the specification, Cascades contends that one is not necessary, because the average computer programmer would understand the structure of a documentation generator. In response, the defendants do not directly argue the contrary, i.e., that a typical software engineer would not understand what the structure of a documentation generator is. Instead, they cite several cases, some of which concern construction of terms including the word "generator" (though none concerns a documentation generator) and others of which concern arguably similar terms, such as "processor." These cases are not directly relevant to the matter at hand. In *Rodriguez*, an administrative decision that is not controlling authority, the Board of Patent Appeals and Interferences invalidated several claims that included the term "system configuration generator." *Ex Parte Rodriguez*, slip op. at 20–27. In that case, unlike this one, the side arguing in favor of the patent's validity had not "ever suggested that these elements are a known structure in the prior art" and in fact "argue[d] that these elements are of their invention and not known in the prior art." *Id.* at 27. Thus, although the defendants label *Rodriguez* "the

best case" on the indefiniteness of "generator" as a term connoting structure, Hrng. Trans. at 82, the facts and arguments there were different.² As *Flo Healthcare* and other Federal Circuit cases on the means-plus-function inquiry make clear, analysis on this question must be case-specific.

The facts were also different in *Aristocrat*, a case the defendants cite multiple times for the proposition that a claim including software must disclose a unique algorithm for that software. There, the disputed terms used the word "means," which meant the court did not have to consider the presumption that section 112(f) does not apply when that word does not appear. *Aristocrat*, 521 F.3d at 1331 (terms were "game control means" and "control means"). In fact, the parties in that case agreed that the terms at issue were means-plus-function terms. *Id.* The court therefore did not have to evaluate a crucial support for that presumption: whether "the claim term is used in common parlance or by persons of skill in the pertinent art to designate structure." *Inventio*, 649 F.3d at 1359–60. Instead, the plaintiff in *Aristocrat* argued that "devising

² The defendants also cite two district court cases that mention "generator" terms. In one, the court did not appear to consider whether the term at issue, "report generator for outputting," would reveal sufficient structure to one of ordinary skill in the field. Indeed, the court did not discuss the presumption against application of section 112(f) when the word "means" does not appear in the claim term. See *Isogon Corp. v. Amdahl Corp.*, 47 F. Supp. 2d 436, 449–50 (S.D.N.Y. 1998). Of course, that case was decided before *Lighting World*, *Inventio AG*, *Flo Healthcare* or any of the other cases discussing the presumption cited here. The defendants cite another district court case predating those decisions, but there, the parties agreed that the term "state generator" was a means-plus-function term, and thus the court did not have to decide whether the presumption applies. See *QSIndustries, Inc. v. Mike's Train House, Inc.*, 230 F. Supp. 2d 1240, 1245–46 (D. Or. 2002). The defendants also cite a "triumvirate of recent decisions" from the Patent Trial and Appeal Board, but each of those cases construed the term "processor," and the defendants make no argument explaining why "processor" and "documentation generator" are equivalent for this purpose. See Defs.' Br. at 14 (citing *Ex Parte Erol*, No. 2011-001143, 2013 WL 1341107, at *9 (P.T.A.B. Mar. 11, 2013); *Ex Parte Lakkala*, No. 2011-001526, 2013 WL 1341108, at *7 (P.T.A.B. Mar. 11, 2013); *Ex Parte Smith*, No. 2012-007631, 2013 WL 1341109, at *8 (P.T.A.B. Mar. 12, 2013)).

an algorithm to perform [the invention's] function would be within the capability of one of skill in the art." *Aristocrat*, 521 F.3d at 1334.

That is not what Cascades is arguing here. Rather, it contends that "those of skill understand that a 'documentation generator' is a structure, i.e., a programming tool." Pl.'s Resp. at 15. To support this argument, Cascades offers definitions of "document" and "generator" from a scientific and technical dictionary; the dictionary notes that "generator" is a computer science term and defines it as a "program that produces specific programs as directed by input parameters." Pl.'s Ex. D at 4. Cascades also cites two Wikipedia pages: one defining "documentation generator," and another providing a list of documentation generators produced by various programmers and companies. See Pl.'s Exs. E, F. The defendants have attacked these sources as dated after the patent application and "'less significant' than the intrinsic evidence," Defs.' Repl. at 10. The Court agrees that these sources are not perfect, but they do provide some support for the proposition that a programmer of ordinary skill would understand the structure of a documentation generator. By contrast, the defendants have not provided evidence that programmers in the field would *not* understand the structure of a documentation generator.

For its part, Cascades also contends that there is at least one example of the structure of a documentation generator in the specification itself, where it describes a "dynamic binary translator" that "can generate host code for every foreign instruction in sequential order with the following properties. Dynamic binary translator uses a subset of host registers in register file 100 (FIG. 1) to map the foreign registers into the host registers." Pl.'s Resp. at 15–16 (quoting Patent at 8:65–9:3). At the hearing, Cascades

argued that this description "is a structure of a documentation generator." Hrng. Trans. at 94. The defendants respond that "dynamic binary translator" is already a claim term (although it is not, strictly speaking; the disputed claim term is "binary translation/translator," without the "dynamic"). Cascades is not arguing, however, that the passage it points to provides *the* structure of a documentation generator; rather, it is "an example of a generic structure," which "extracts information from the host code and maps foreign registers into the host registers for creating documentation." Pl.'s Resp. at 15. In addition, elsewhere in the specification, the patentees wrote that "[t]he documentation is created during the optimizing binary translation processes 202," pointing to one of the diagrams in the patent. Patent at 15:17–18.

The Court cannot hold the "documentation generator" term invalid for indefiniteness without evidence from the defendants that computer programmers would not understand the structure of Claim 1's documentation generator in the context of the whole patent. This conclusion is supported by the Federal Circuit's admonition that the presumption against construing a claim term without "means" as means-plus-function term "is a strong one that is not readily overcome." *Lighting World*, 382 F.3d at 1358. Cascades has invoked this presumption, and it has pointed to an example of a structure for a documentation generator in the patent itself. In response, the defendants have pointed to inapposite cases that do not involve the term "documentation generator" and in which the patent uses the word "means" or the parties agreed that a particular term was a means-plus-function term. For all of these reasons, the Court agrees with Cascades that this term is not subject to section 112(f), and that its definition is in line

with what Cascades says a typical computer programmer would understand a documentation generator to be: a programming tool that produces documentation.

6. "Operations required to calculate a corresponding foreign state"

The term "operations required to calculate a corresponding foreign state" appears in both Claim 1 and Claim 15. In Claim 1, it is included in the context of the "documentation tracker," which is "configured to record host operation addresses at appointed points of the host operation sequences being executed, wherein, for each host operation address, operations required to calculate a corresponding foreign state for the host operation address are added to documentation." Patent at 16:20–25. The term is similarly situated within Claim 15, although it is in reference to a "recovery point" and not "appointed points." Claim 15 describes a recomputing method that generates a "set of documentations," "wherein each documentation in the set of documentations corresponds to a recovery point in the optimized binary translated code and describes operations required to calculate a corresponding foreign state for the recovery point." *Id.* at 17:29–34.

Defendants' proposed definition includes the entirety of the term at issue, but then adds this clause: "but not by 'roll[ing] back its target registers to recover state.'" See Joint Notice of Modified Proposed Claim Constructions at 2 [docket no. 123]. They say this limitation comes from the patent's prosecution history, where, defendants argue, "the patentee made a definitive statement that distinguished its 'operations required to calculate a corresponding foreign state' from the prior art's way of 'roll[ing] back its target registers to recover state.'" Defs.' Br. at 16. Specifically, defendants argue that the patent examiner rejected some of the claims that did not make it into the

final patent, because they were "anticipated by" another patent, the Kelly patent. See Defs.' Ex. 2 at 84–87, 92–94. Cascades, as elsewhere, advocates for use of the plain and ordinary meaning of this phrase rather construing it, but alternatively proposes this definition: "the actions resulting from instructions required to determine a corresponding non-native state." See Joint Notice of Modified Proposed Claim Constructions at 2–3 [docket no. 123].

The Kelly patent used an operation called a "rollback" when handling exceptions, and defendants say the patent examiner initially rejected what is now Claim 1 of the '750 patent because of Kelly. At the time, the defendants argue, the '750 patent did not include the phrase at issue here ("operations required to calculate a corresponding foreign state"); they say the patentees added it after the rejection. "[O]n the basis of that amendment, [patentees] argued that the claims were novel over Kelly and Kelly's rollback feature described above." Defs.' Br. at 17. The defendants cite the applicants' statement to the Patent Office contrasting the Kelly patent based specifically on the rollback language, which they argue is used in context of the "operations required to calculate a corresponding foreign state" term. *Id.* (citing Defs.' Ex. 2 at 75.) Defendants argue that the definition of this term therefore must include the limitation that distinguishes the '750 patent from Kelly.

At the hearing, Cascades argued that "in fact, the patentee never disclaimed the rollback operation." Hrng. Trans. at 102. They argued that "[i]nstead, the patentee said that just because Kelly has this rollback operation does not mean that documentation is inherently disclosed in Kelly." *Id.* at 104. As support, Cascades cites the same page of the letter just referenced, in which the applicants stated that "it is not inherent in Kelly to

store a set of documentations each including operations for host operation addresses" Def.'s Ex. 2 at 75. Cascades argued at the hearing that "[t]he patentee never said that if there is documentation . . . and the described operations, then there can be no rollback. He simply stated that Kelly does not disclose the documentations, therefore, it's distinguishable." Hrng. Trans. at 105. In its brief, Cascades also contends that negative limitations like the one defendants propose are disfavored by the Federal Circuit without "a firm anchor in the specific claim language of the specification of the patent." Pl.'s Resp. at 17 (citing *Linear Tech. Corp. v. ITC*, 566 F.3d 1049, 1060 (Fed. Cir. 2009)).

There is a "heavy presumption that claim terms carry their full ordinary and customary meaning, unless [the defendant] can show the patentee expressly relinquished claim scope" during the prosecution of the patent. *Epistar Corp. v. Int'l Trade Comm'n*, 566 F.3d 1321, 1334 (Fed. Cir. 2009). The alleged disavowal must "be both so clear as to show reasonable clarity and deliberateness, and so unmistakable as to be unambiguous evidence of disclaimer." *Omega Eng'g, Inc. v. Raytek Corp.*, 334 F.3d 1314, 1325 (Fed. Cir. 2003) (citation omitted). The Federal Circuit has "consistently rejected prosecution statements too vague or ambiguous to qualify as a disavowal of claim scope." *Id.* (collecting cases)). To qualify, though a "disavowal of claim scope" can be made "through arguments made to distinguish prior art references," these arguments must "constitute clear and unmistakable surrenders of subject matter." *Cordis Corp v. Medtronic Ave, Inc.*, 511 F.3d 1157, 1177 (Fed. Cir. 2008).

The disputed page of the prosecution history is part of a December 2005 submission by applicants to the patent examiner supporting what eventually became the

'750 patent. The examiner had found claim 35 (which ended up as claim 1 of the '750 patent) to be anticipated by the Kelly patent. The examiner noted that although the Kelly patent did not explicitly disclose a "documentation tracker configured to track an executable path" to be used when an exception occurs, it did disclose "code morphing software that handle exceptions and errors . . . by replacing working state with correct target state as necessary." Defs.' Ex. 2 at 87. The examiner stated that "[i]t is inherent that some type of 'documentation' is kept by the executing system to track successful / unsuccessful execution of code segments, used to evaluate and correct binary translations." *Id.*

In response, the applicants amended claim 35 in an effort to address the examiner's finding. Among other things, they added the language currently in question: "operations required to calculate a corresponding foreign state for the host operation address are added to documentation." *Id.* at 69.

Together with their amendments, the applicants offered a narrative submission in which they sought to distinguish the claim, as amended, from the Kelly patent. The applicants first described how the Kelly patent deals with exceptions: "Kelly discloses transferring the contents of all working registers to official target registers and allowing an operation called a rollback to quickly transfer the content of all official target registers back to their working register equivalents." Defs.' Ex. 2 at 75. "If an exception occurs [in the Kelly invention], the original state in the target registers at the last update (or commitment) may be recalled to the working registers." *Id.* The applicants then contrasted their claim: "In contrast, claim 35 'discloses adding operations required to calculate a corresponding foreign state for the host operation address are added [sic] to

documentation." *Id.* (emphasis in original). If an exception occurs, the operations are executed to recover a foreign state and then continue execution of the foreign code. *Id.*

The applicants contended in their submission to the examiner that Kelly was different, because it "discloses rolling back contents of official target registers back to the working registers' equivalents Kelly does not need to use documentation to perform a set of operations because Kelly rolls back its target registers to recover state." *Id.* The applicants also addressed the examiner's statement that it was "inherent" that Kelly used documentation to allow operations to continue: they stated that it was not inherent in Kelly to store a set of documentations and then to determine a documentation to recover a foreign state, again, "because Kelly rolls back its target registers to recover state." *Id.*

It is clear that the patent applicants sought to differentiate their invention from Kelly by pointing out that Kelly does not use documentation to perform recovery operations after an exception because it rolls back its target registers to "recover state." The question, however, is whether this amounts to a clear and unequivocal statement that the reference in the '750 patent to "operations required to calculate a corresponding foreign state" *excludes* rolling back the target registers. The Court concludes that the patentees did, in fact, clearly and unmistakably disclaim the use of a rollback function. They did so by their statements to the patent examiner in which they expressly differentiated their patent's methodology from the rollback function employed by the Kelly patent. In essence, the patentees said that how the Kelly patent operated in this regard and how their patent operated were two distinct ways of accomplishing a particular goal. Because the patentees differentiated the two patents in order to

distinguish a prior art reference that otherwise would have invalidated their patent, they clearly and unmistakably surrendered the use of a rollback functionality.

For these reasons, the Court adopts defendants' proposed limitation on this claim term. The "operations required to calculate a corresponding foreign state" as described in claim 1 do not include rolling back the target registers to recover an earlier state.

7. "Precisely"

The word "precisely" appears just once in the claims at issue in this case. In Claim 1 of the patent, there is description of a "binary translation system." The first element of that system is "a non-optimizing foreign code execution module configured to maintain dedicated foreign state for each foreign binary operation executed allowing for the exceptions arisen to be handled precisely." Patent at 16:6–9.

Defendants propose a detailed definition for this seemingly commonplace word: "a later instruction has not yet been executed and all prior instructions preceding the instruction causing the exception have executed and committed their results." Defs.' Br. at 18. That passage comes from a section of the specification describing an example of how the system works, dealing with hypothetical instructions "ie" and "im": "The exception generated by instruction 'ie' is a 'precise' exception if instruction 'im' has not yet been executed and all prior instructions preceding instruction 'ie' have executed and committed their results." Patent at 8:26–29. Defendants contend that the inventors acted as their own lexicographer for the term "precise" by writing this passage, because "[t]he patent takes great effort to explain exactly what 'precise' means in the context of its invention." Defs.' Repl. at 13.

As with other terms, Cascades argues that "a person having ordinary skill in the art at the time of the invention would understand 'precisely' to have its plain and ordinary meaning." Pl.'s Resp. at 21. It says the term "describe[s] how exceptions *are handled precisely*"—a concept that Cascades says is different from a "precise exception," which it says the defendants draw from for their definition. *Id.* at 19. "[A]n exception is 'handled precisely' when the behavior of foreign code, including precise exceptions, is maintained." *Id.* (This is apparently what Cascades means by "plain meaning.") Cascades then argues, without quoting the patent, that "[p]recise exceptions can be defined as exceptions when running binary translated code that are processed by the host system to correctly emulate the foreign architecture." *Id.* at 20. Cascades also contends that defendants draw their definition from a part of the patent that is discussing "speculative mode," which it argues is "not the proper definition of 'precisely.'" *Id.* at 21. At the claim construction hearing, Cascades argued that the plain meaning of "precisely" is adequate, because the term is "quite well-described in Claim 1." Hrng. Trans. at 108.

One difficulty in the use of "precisely" in Claim 1 is that the term does not appear anywhere else in the patent. However, the term "precise" appears many other times, including in the title of the patent: "Method and Apparatus for Preserving Precise Exceptions in Binary Translated Code." The specification often refers to the importance of "maintaining precise exceptions," and "[t]he ability to preserve the behavior of the foreign code, including precise exceptions." Patent at 3:31, 23–24. "[I]t is desirable to support precise exceptions in a host architecture in a manner that correctly emulates a foreign architecture." *Id.* at 3:50–52. Further, the specification describes "[t]he present

invention" as a way to execute binary translated code "in a manner . . . that supports precise exception maintenance." *Id.* at 3:56–59. Although Cascades appears to argue that a "precise exception" and an exception that is "handled precisely" are different things, there is not one instance in the patent in which "precise" is used as anything but a modifier for the word "exception." It would not appear to be a stretch to conclude that "precise exception maintenance" and handling an exception "precisely" are similar concepts, and perhaps the same concept.

For these reasons, adoption of the defendants' proposed definition appears appropriate. Though it comes from a definition of "precise" in the context of an exception, Cascades does not present any good reason why a definition for the term "precisely" cannot be drawn from the patent's use of the term "precise." The defendants' definition does come directly from the patent, specifically, from a passage where the patent declares when an exception "is" precise: "The exception generated by instruction 'ie' is a 'precise' exception if instruction 'im' has not yet been executed and all prior instructions preceding instruction 'ie' have executed and committed their results." Patent at 3:8–12. The Cascades definition, that "precisely" means the behavior of the foreign code is maintained, has a few disadvantages. Though the notion of maintaining a foreign state appears in the same clause of Claim 1, that clause notes that handling exceptions precisely is something *allowed by* maintaining a dedicated foreign state. See *id.* at 16:6–9. That is not the same thing as defining "precisely." Further, defendants' definition is more comprehensive, in that it specifically describes what must happen for an exception to be considered precise.

The Court therefore adopts the defendants' construction of "precisely," but adds the words "such that" at the beginning of the definition in order to make the interpretation grammatically correct; "precisely" is an adverb. That construction is: "such that a later instruction has not yet been executed and all prior instructions preceding the instruction causing the exception have executed and committed their results."

8. "Recovery point"

The patent references "recovery points" several times in Claim 15, though not at all in Claim 1. Claim 15 describes a recomputing method; among its functions is "designating a set of recovery points in the optimized binary translated code during optimized translation of the foreign code, wherein each recovery point represents a foreign state." Patent at 17:25–23. Recovery points are also discussed in conjunction with documentation: "each documentation in the set of documentations corresponds to a recovery point in the optimized binary translated code and describes operations required to calculate a corresponding foreign state for the recovery point." *Id.* at 17:30–34. The claim then goes into detail about how a documentation and a recovery point work together: a documentation "correspond[s] to executed optimized binary translated code when an exception arises during its execution to recover a foreign state corresponding to a recovery point for the exception, wherein the foreign state is recovered by executing the operations for the one of the documentations." *Id.* at 18:2–7.

Here again, the defendants argue that their definition of the term is expressly laid out in the patent itself. They point to a passage of the specification stating that "a set of

'Recovery Points' (RP) are [sic] included in the binary translated code, with the following properties." Patent at 9:30–32. This passage then lists three properties, each of which defendants truncate and include in their definition of this term: "a point in the binary translated code that (1) has a correspondence with an instruction in the foreign code; (2) is described by a documentation and (3) any synchronous exception between adjacent recovery points can be reinvoked by interpreting foreign instructions starting from the previous recovery point." Defs.' Br. at 20. Defendants then argue that each of these properties is also reflected in Claim 15.

Cascades says the meaning of this term would be apparent to one of ordinary skill in this field. However, Cascades alternatively defines "recovery" with a dictionary definition: "restoration or return to a former[,] usual[,] or correct state or condition." Resp. at 22 (citing Pl.'s Ex. H (commas restored; Cascades omitted them)). Cascades also argues that, if plain meaning is not acceptable, "restoration point" is an appropriate definition "because a recovery point seeks to bring the host program configuration back to a correct dedicated foreign state in the case of an exception." *Id.* Cascades quotes two passages from the specification, one stating which type of information is saved at recovery points, and another defining what a "recovery mechanism" does. See Patent at 4:60–65, 17:16–19. From these passages, Cascades concludes that "[a] recovery point is thus a point at which saved information can be restored to assist the recovery mechanism." Resp. at 22. Further, Cascades says the defendants' definition "misstates the claims," citing one example: though defendants' definition states that "[e]very Recovery Point is described by a documentation set," Claim 15 says "each documentation in the set of documentations *corresponds to* a recovery point." Resp. at

22 (quoting Patent at 17:16–19 (emphasis added by Cascades)). At the claim construction hearing, Cascades emphasized that the defendants' construction is "not a definition," because it lists "features of a recovery point" but does not define it. Hrng. Trans. at 111.

Defendants reply that the Cascades definition is a "mere substitution of the word 'restoration' for 'recovery,'" and that only defendants' definition relies on actual language from the patent. Defs.' Repl. at 15. At the hearing, they observed that the patent refers to "Recovery Points" and that they draw their definition from a passage of the patent that similarly capitalizes the first letter in "Recovery" and in "Point." Defendants contended that the patentees were "consistent in referring back to the recovery point" in the way in which it is described in the defendants' definition. Hrng. Trans. at 110–111.

As with some other terms analyzed above, the difficulty with the defendants' definition is that none of the three features the definition ascribes to a recovery point actually states what a recovery point *is*. Parts one and two of the definition do not describe the nature of a recovery point, but rather say that a recovery point "has a correspondence with an instruction in the foreign code" and "is described by a documentation." No knowledge is gained of the actual nature of a recovery point from these descriptions. Part three is even less of a definition: "any synchronous exception between adjacent recovery points can be reinvoked by interpreting foreign instructions starting from the previous recovery point." Cascades, on the other hand, accomplishes little with its own definition ("restoration point"), in that it merely swaps out one word for another with little change in meaning.

The term "recovery point" is used with equal applicability in multiple situations in the patent. For example, in the "Summary of the Invention" section, the patent states that "state information is saved at a plurality of recovery points in the binary translated code" during the translation process. Patent at 4:60–61. At another point, the specification states that "[w]ith the information provided by documentation 528, exception handler 532 is able to recreate the state of the host computer system at the most recently executed recovery point." *Id.* at 15:25–28. Not all of these mentions of recovery points put the term in capital letters, contradicting the defendants' argument that the patentees intended an official, exclusive definition for the term. In fact, they point to a general, identifiable concept that is amenable to a simpler construction.

Although Cascades' proposed definition of "restoration point" is not helpful, Cascades goes further in its brief toward actually defining what a recovery point *is*. There, it included two passages from the patent discussing recovery points to conclude that a "recovery point is thus a point at which saved information can be restored to assist the recovery mechanism." Pl.'s Resp. at 22. Likewise, during their portion of the tutorial at the hearing, the defendants said that "a recovery point is set in the patent every time we're running an instruction that might cause an exception. So every time we're concerned we might be doing something that might go wrong, we set that as a recovery point." Hrng. Trans. at 35–36. (Elsewhere, the defendants described a recovery point as "the last known point where everything was okay." *Id.* at 36.) These statements are much clearer about the nature of a recovery point and the function it serves in the greater patent scheme. The Court therefore construes the term "recovery

point" to mean a "location in binary translated code that is marked prior to the occurrence of an exception."

Conclusion

The Court rules on the disputed claim terms as set forth in the foregoing decision. These cases are set for a further status hearing before the undersigned judge on January 21, 2014 at 9:30 a.m. The parties are directed to confer prior to that date regarding whether there are additional common issues that might warrant expansion of the IOP 13 pretrial consolidation and are to submit a status report containing their positions in this regard by no later than January 17, 2014. The status report should include each party's positions on what, if any, further fact discovery is appropriate under Local Patent Rule 1.3 as well as each party's intentions with regard to the disclosure of expert witnesses.



MATTHEW F. KENNELLY
United States District Judge

Date: January 2, 2013